August
'84

Das Magazin für den Home-Computer-Besitzer

computronic

8/1. Jahrgang

DM 5,50
öS 45
SFR 5,50

Commodore 64
ZX-Spectrum
ZX-81
Dragon 32/64
Atari
VC-20
TI-99
Apple II
Laser 2001

Forth-Kurs
Teil 5

Kassetten-
Blitzservice

SUPER Programme
Maya
Andromeda
Monster-Attack
Painter
Solitaire
Zyklo
und viele, viele mehr

Neu! Jetzt auch Programme für den LASER 2001

Möchten Sie als Software-Autor für den TRONIC-Verlag tätig werden?

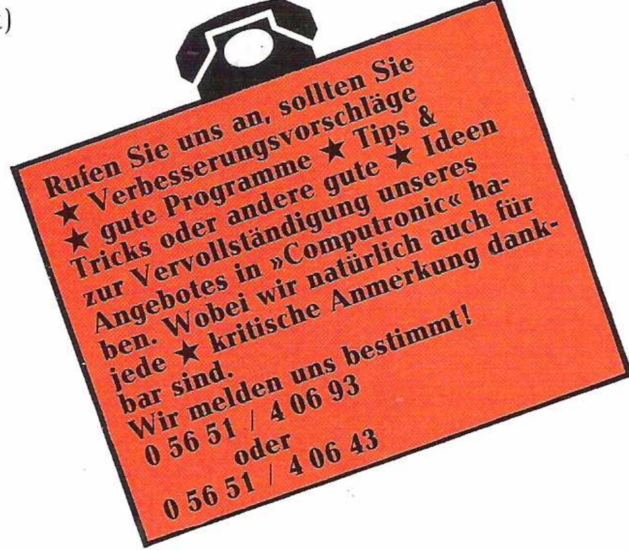
Wir würden uns freuen, in Ihnen einen Ansprechpartner für eine künftige intensive Zusammenarbeit zu finden. Der TRONIC-Verlag vereinbart mit seinen künftigen Software-Autoren pro veröffentlichte Seite (einschl. Programmbeschreibung) ein Honorar von **DM 120,–.**

Dieses Entgelt wird fällig, wenn die Redaktion des Verlages sich für eine Veröffentlichung entscheidet. Die Auszahlung erfolgt also nicht erst nach Veröffentlichung in einer unserer Ausgaben, sondern früher.

Der Verlag wird vom Autor berechtigt, seine Manuskripte (Programme) zur Darstellung im Heft heranzuziehen und abzudrucken.

Einzusenden sind:

- Programmbeschreibung
- bespielte Cassette oder Diskette
- Listing (mit Copyright)
- Freiumschlag



Haben Sie Interesse? Haben Sie noch Fragen?

Setzen Sie sich telefonisch mit unserer Redaktion in Verbindung!

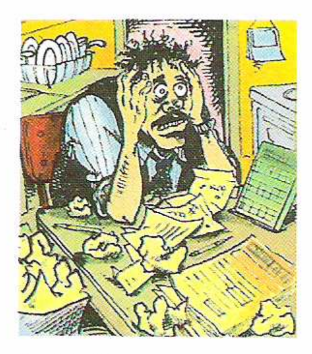
TRONIC-VERLAG

DIE REDAKTION

Aus dem Inhalt:



Ein interessantes Spiel für alle TI-99-Freunde. Mit toller Graphik. Ein Spiel in 378 Bildern



Forth-Kurs

Teil 5 in Computronic. Diesmal: Bedingte Verzweigungen



Laser 2001

Neu – in Computronic! »Andromeda«, ein Programm für den Laser 2001

Im Test der BIT 90

Ein neues Gerät auf dem Computermarkt. Wir haben ihn für Sie getestet. Lesen Sie hierzu Seite 5

In diesem Heft:

8/'84

Software

Aktuelles

Internationale Computer Show '84 Köln

Mit einer Beteiligung von 221 Firmen aus 14 Ländern, darunter alle Branchenführer, erwies sich die C '84 auf Anhieb als Erfolg. Neben 158 deutschen Firmen und einem Unternehmen aus der DDR beteiligten sich 62 ausländische Unternehmen, an der Spitze die USA mit 26 Ausstellern, gefolgt von Großbritannien, Taiwan, Japan und Italien.

Die auf Kleincomputer für Beruf, Heim und Hobby spezialisierte Veranstaltung ist eine Nachfolgeveranstaltung der erfolgreichen US Computer Show, die im vergangenen Jahr erstmals von der US-Botschaft in Zusammenarbeit mit der Köln-Messe durchgeführt wurde.

Die C '84 bot auf überschaubarer Fläche (über 16 000 m²) eine umfassende Übersicht über den gesamten Kleincomputer-Markt. Über Standort-Programme und branchenorientierte Problemlösungen, über Zubehör aller Art sowie Computer- und Videospiele. Sonderveranstaltungen für Besucher der Computer-Show vertieften das Informationsangebot der Aussteller, denn der Markt für Mikrocomputer wird immer unübersichtlicher, vor allem das Softwarenangebot. In der Bundesrepublik kämpfen etwa 250 Hardwareanbieter und rund 600 Softwarenhäuser um Marktanteile. Kundendienst und Ersatzteile lassen darüber hinaus in zahlreichen Fällen zu wünschen übrig.

Neuheiten auf dem Computersektor gab es auf der C '84 nur im Softwarebereich. Dichte Menschenmassen strebten durch die Hallen und umlagerten die Stände der Aussteller. Aber wenn König Fußball kam, war auch der Stand vom Computerclub des WDR leer. Nur vor den Fernsehern standen noch Leute.

Ein Computer-Adreßbuch vom ROM Verlag aus Ludwigshafen wurde auf der C '84 vorgestellt. In diesem Buch kann sich jeder eintragen lassen, der etwas über Computer sucht, tauschen will oder eigene Programme abzugeben hat. Das Buch soll zum erstenmal im IV. Quartal 84 erscheinen.

Jetzt hören schon Computer auf die Stimme ihres Herrn!

Ob chinesisch oder im Kölschen Dialekt programmiert – 450 Begriffe versteht das Wunderding von APPLE. Sollte allerdings ein Fremder befehlen: „Brauche Umsatz von der letzten Maiwoche", hat er Pech: Das Elektronen-Gehirn reagiert nur auf die Stimme dessen, der es auch selbst programmiert hat.

Sehen und ausprobieren konnte man die Neuheiten auf der Messe. Gezeigt wurden unter anderem Computer, die auf Fingerzeig auf den Bildschirm reagieren, transportable Mini-Computer für unterwegs, die auch an eine 12-Volt-Autobatterie angeschlossen werden können.

Weg von der Gewalt, unter diesem Motto hat die Firma TOBY-DISK einige Programme für Kinder herausgegeben. Diese Programme befassen sich mit den Märchen aus Tausendundeiner Nacht. (Hotel der Tiere, Fliegender Teppich usw.). Eben Programme wie sie keiner kennt – Technik – Spiel – Musik – Zeichnen – Rechnen – leicht gemacht.

Als Eltern ihre Kinder bei TOBY-DISK wieder abholen wollten, gab es Geschrei und Gezeter, denn diese wollten einfach nicht mit nach Hause. Sollten Sie Probleme mit Staub haben, die Firma ERNO bläst alles weg. Auf dem Fotosektor ist diese Firma schon lange bekannt und nun bietet sie Dust-Off-Pflegemittel auch für Computer an. Ein Druck auf den Knopf und schon ist der Staub auf Disketten und Computer weg.

Ein neues Diskettenlaufwerk unter der Bezeichnung „MICROPOWER 2000" ist von der Firma RMC aus Oberhausen auf der Messe vorgestellt worden. Die Flobby-Disk verfügt über 2 Laufwerke und kann ohne Interface an alle COMMODORE-Rechner angeschlossen werden. Das Gerät ist schreib- und lesekompatibel mit dem 1541 und CBM 4040. Mit dem Diskettenlaufwerk können in 2 Minuten Disketten dupliziert werden. Zum Preis von ca. 2700,– DM ist die MICROPOWER 2000 im Fachhandel erhältlich. Ein Test ist in Vorbereitung.

Zusammenfassung: Alles in allem eine Computershow der Superklasse, die



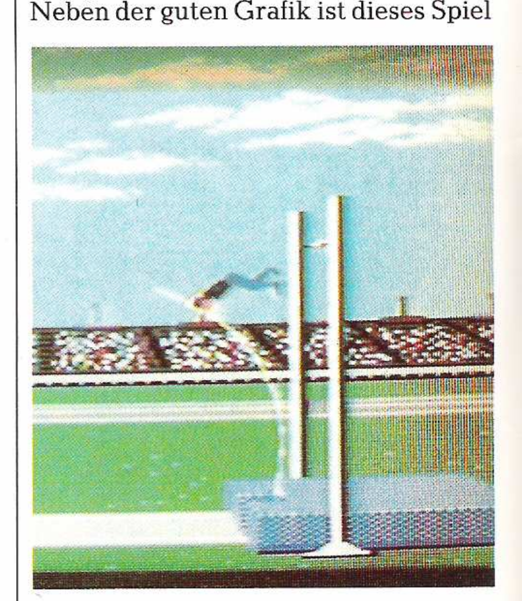
leider unter dem Fußball zu leiden hatte. Den nächsten Termin sollten Sie sich schon jetzt merken und aufschreiben. Die 2. Internationale Computershow Köln ist vom 13.–16. Juni 1985. -

Rolf Freitag

Software

Summer Games

Was für den Sportler die Olympischen Spiele in Los Angeles sind, ist für den Computerfan Summer Games.
Neben der guten Grafik ist dieses Spiel



ein Spiel der Superklasse. Nachdem das Programm in den Rechner geladen ist, kommt die Eröffnungsfeier mit dem Anzünden des Olympischen Feuers. Danach starten Friedenstauben aus dem Stadion. Jetzt kommt die Aufforderung seinen Namen einzugeben und unter 18 Länder kann man seine Nation heraussuchen. Das Spiel ist für 1–8 Personen geeignet. Nach Abspielen der Hymnen kommt ein Auswahlmenü, in dem man sich entscheiden kann, alle 8 Optionen der Reihe nach mit Wertung oder ohne Wertung zu spielen. Es besteht auch die Möglichkeit, jede der einzelnen Sportarten anzuwählen.
Die Sportarten im einzelnen: Stabhochsprung, Turmspringen, 4x400-m-Staffellauf, 100-m-Lauf, Gymnastik, Freistilstaffel, 100-m-Freistil und Tontaubenschießen. Nach Beendigung der Durchgänge erklingt für den Sieger die Nationalhymne und ein Punktestand. Das Spiel ist bei der Firma SOFTSELL in München zum Preis von ca. 117,– DM erhältlich. RF

Fachtagung in Berlin

Vom 8.–10. Oktober 1984 findet an der Freien Universität Berlin die Fachtagung der Gesellschaft für Informatik „Informatik – Herausforderung an Schule und Ausbildung" statt. Ziel der Tagung soll sein, die Ausbildung in Informatik, ihre Auswirkungen und Realisierungen in den verschiedenen Bildungs- und Berufsbereichen darzustellen, und Lehrer, Wissenschaftler, Dozenten, Lehramtsstudenten, Behörden und Verwaltungen sowie Ausbildungsleiter der Industrie zu Gesprächen zusammenzuführen.

Im Test der BIT 90

Beim Auspacken des Computers wurden wir von der Vielzahl der Kabel überrascht. Neben dem Netzteil und dem Antennenkabel war ein Kabel für den Kassettenrecorder und einem zusätzlichen Monitor vorhanden. Dem BIT 90 ist auch noch eine Antennenweiche beigelegt. Diese ermöglicht eine Umschaltung zwischen Hausantenne und Computer. So fällt das Umstecken der Antennenkabel flach.

Nach dem Anschließen des BIT 90, das im Handbuch durch Zeichnungen unterstützt wurde, wußten wir nicht, ist der BIT 90 nun in Betrieb oder nicht, denn es fehlt die Kontrollampe. Nach Aussage des Händlers wird dieses jedoch abgeändert. Mit seinem silberfarbenen Gehäuse, der grünen Tastatur auf schwarzem Grund und flachen Design hat der BIT 90 ein Aussehen, das ins Auge fällt.

Die Tastatur besteht aus 66 Gummitasten, wie beim Sinclair oder Laser, wovon manche Tasten bis zu viermal belegt sind. Die Funktionen der Tastenbelegung ist über vorgegebenen Tasten einfach und für jeden erlernbar.
Alle BASIC-Befehle, die der BIT 90 kennt, sind über die BASIC-Taste erreichbar. Die Kürzel müssen nicht benutzt werden, der BIT 90 versteht auch normale Eingabe. Auf der rechten Seite hat der BIT 90 die 4 Cursortasten, die ein freies Bewegen auf dem Bildschirm ermöglichen sollen. Aber weit gefehlt, eine Änderung ist nur in der letzten Zeile möglich, vor Drücken der Returntaste. Sicherlich ein Nachteil, aber über >Edit< ist ein freies Bewegen auf dem Bildschirm möglich. Der Programmierer, der seine Programme ändern will, muß mit >Edit< arbeiten. Dieses wird mit zwei Tasten unterstützt.
Ernsthaften Programmierern steht nichts im Wege – der >Made in Taiwan<-Computer bringt sogar eine Reihe Besonderheiten mit, die dem Anwender das Arbeiten leichter machen. Nicht nur die genannte Kurzwahl der BASIC-Befehle per Tastendruck, sondern auch der „AUTO-Modus" und der „REMUN-Modus", die eine automatische Durchnumerierung der Programmzeilen in wählbarer Schrittweite übernimmt.
Die 4 Cursortasten können einen Joysick ersetzen, so daß der Spieler eine einfache Handhabung hat und zudem auch noch Geld spart. Der BIT 90 besitzt 2 Joystickspots, wo sich jeder handelsübliche Joystick anschließen läßt. In einer Programmzeile finden 128 Zeichen Platz und es bereitet keine Schwierigkeiten, ein weiteres Programm auf ein bereits eingespeichertes Programm aufzuladen.
Der BIT 90 besitzt Befehle, die andere Computer erst durch teuere Programme wie Simon s Basic für den Commodore oder EX-Basic wie Trace, Else, Inky usw., erhalten.
Durch einen einfachen Tastendruck „COPY" erhalten wir einen Hardcopyausdruck auf unsern Drucker. Es läßt sich fast jeder Drucker ohne Probleme aufgrund der RS 232 C Schnittstelle anschließen.



Mit einer 256x192-Grafikauflösung ist eine sehr gute grafische Gestaltung in 16 Farben möglich. Die grafische Darstellung des BIT 90 liegt deutlich über dem Durchschnitt.
Auf Verlangen gibt der BIT 90 auch Töne von sich. DreiTon- und Rauschgeneratoren mit fünf Oktaven sorgen für fast jedes Geräusch. Damit bringt der BIT 90 die besten Voraussetzungen für Spiele mit sich.
Über Software braucht sich der Anwender nicht zu beklagen. Auf dem Spielsektor sind schon Programme zu haben. Nur im Anwenderbereich ist noch kein großes Angebot vorhanden. Da der Computer jedoch auch Atari und Colecovision-Programme verarbeitet, besteht auch hier eine große Auswahl an Programmen zu Verfügung.
Jeder Kassettenrecorder ist anschließbar. Leicht zu vertauschen ist der Kassettenrecorderanschluß mit den Joystickbuchsen, da diese die gleiche Buchse haben. Der BIT 90 ist eine Mischung aus APPLE und TSR-80.. Außer dem umfangreichen Basicinterpreter besitzt der BIT 90 noch folgende Hardwarespezifikationen:

X CPU Z-80A, 3,58 MHz
X ROM 24 K
X RAM 34 K
X RAM erweiterbar auf 50 K
X 66 Standardtasten
X Groß- und Kleinschreibung
X Grafikmode
X 4 Tongeneratoren
X 16 Farben
X 32 Sprites
X 2 Joystickbuchsen
X Kassetteninterface mit 2400 Baud Bit sx Serielle RS 232 C Schnittstelle

Fazit:
Zum Erstellen von größeren Programmen ist unserer Meinung nach die Gummitastatur nicht geeignet. Das mag Gewöhnungssache sein, die meisten Benutzer ziehen eine solide Schreibmaschinentastatur jedoch vor. Die fehlende Kontrollampe wird eingebaut. auch der Kassettenanschluß wird überarbeitet.

Alles in allem, der BIT 90 bietet ein gutes Preis-/Leistungsverhältnis im Computerbereich und ausreichend Software.

Wenn der BIT 90 wirklich ein Volltreffer werden sollte – wir wünschen es dem BIT 90 –, wird die Literatur nicht lange auf sich warten lassen. Der BIT 90 wäre es wert. RF

Erfolgreiche Durchführung des 4. HESSISCHEN COMPUTER-TAGES

Am 5. 12. 1982 fand zum ersten Mal eine Veranstaltung statt, die es in dieser Form noch nicht gab. Um diese Art der Veranstaltung von den bekannten großen Messen zu unterscheiden, wurde vom Veranstalter ein neuer Begriff geprägt: COMPUTER-TAG. Dieser Begriff ist zwischenzeitlich schon zum Bestandteil der deutschen Sprache geworden: Fachzeitschriften benutzen ihn in Überschriften und eine ganze Reihe inzwischen auf dem Markt befindlicher Aussteller, haben diesen Begriff übernommen und teilweise die Veranstaltung bis ins Detail kopiert. Dies zeigt, daß der Veranstalter mit seinen COMPUTER-TAGEN einen Trend stark beeinflußte, wenn nicht sogar begründete: die regionalen Computer-Verkaufsausstellungen.
Der Erfolg dieser Veranstaltungen zeigt sich an den Besucher- und Ausstellungszahlen des 4. HESSISCHEN COMPUTER-TAGES. Mehr als 5500 Besucher und über 50 Firmen ließen diesen COMPUTER-TAG zu einem glänzenden Erfolg weden. Haben beim 1. HESSISCHEN COMPUTER-TAG noch die Aussteller von Home- und Spielcomputern überwogen, zeigten die Aussteller auf dem 4. HESSISCHEN COMPUTER-TAG überwiegend professionelle Anwendungen.

Anzumerken ist, daß auch das Publikum diese Entwicklung befürwortet. Der Besucher, der Lösungen zu Bewältigung seiner betrieblichen Probleme suchte, überwog.

Eine wesentliche Veränderung wird es zum 5. HESSISCHEN COMPUTER-TAG geben. Die Veranstaltung findet dann zum ersten Male an zwei Tagen, dem 8. und 9. Dezember 1984 in Neu-Isenburg statt.

C 64

Listschutz

In der Juni-Ausgabe haben wir bereits eine List-Schutz-Routine vorgestellt. Leider hat sich erst später herausgestellt, daß einige Basic-Befehle nach dem Schutz nicht mehr ausgeführt werden. Dies kommt durch die Arbeitsweise des Basic-Interpreters und kann nicht verhindert werden.
Die nun hier vorgestellte Schutz-Routine ist etwas arbeitsaufwendiger, dafür jedoch in fast allen Zeilen zu verwenden. Der Vorteil ist, daß sich auch einzelne Zeilen schützen lassen. Beim Listen von geschützten Zeilen erscheint nur die Zeilen-Nummer.
Und so wird's gemacht:
Vor jede zu schützende Zeile wird 5 mal die eckige Klammer gesetzt (hinter Zeilennummer und vor Befehl).
Beispiel:
30[[[[[PRINT „DIESE ZEILE IST GESCHUETZT"
Die zu schützenden Basic-Programme sollten mit einer Zeilennummer >5 beginnen, da in den ersten Zeilen die Schutz-Routine eingetragen wird.
Schutzroutine:

```
1 N=2049
2 IF PEEK(N+4)=ASC("[") THEN POKE N+4,0
3 N= PEEK(N)+256 · PEEK(N+1)
4 IF N>0 THEN 2
5 END
```

Durch den Befehl „RUN" werden nun die gekennzeichneten Zeilen geschützt. Die Befehle GET, INPUT und DEFFN dürfen nicht geschützt werden.

2 × Spectrum

Ein Scroll wie beim ZX-81 kann man in Programmen durch >RANDOMIZE USR 3280< ersetzen.

Die fre (0) Funktion fehlt beim ZX-Spectrum. Man kann die noch freien Bytes folgendermaßen erfahren: Print 65536 – USR 7962

24-Zeilen-COPY für den Sinclair ZX Spectrum

Wie man die Bildschirmzeilen 22 und 23 nutzen kann, stand bereits in verschiedenen Fachzeitschriften und soll daher hier nur kurz erwähnt werden:
Zeile 22: PRINT O; AT O,...
Zeile 23: PRINT O; AT O,...
Schwieriger wird es dagegen, wenn man den Bildschirm einschließlich der Zeilen 22 und 23 auf dem Drucker ausgeben will. Die COPY-Routine des Spectrum-ROM verarbeitet nämlich nur die 22 Zeilen des oberen Bildschirmteils.
Abhilfe schafft hier ein kleines Programm, das die beiden Zeilen des unteren Bildschirmteils nacheinander in den Printerbuffer kopiert und dann mit LPRINT auf dem Drucker ausgibt.
Das eigentliche Programm zum Kopieren und Ausdrucken der Zeilen steht in den Programmzeilen 110 bis 140. Es sieht auf den ersten Blick vielleicht etwas kompliziert aus, aber es hat den Vorteil, daß es ohne Veränderung mit beliebigen Zeilennummern versehen werden kann. Außerdem spielt der unterschiedliche Aufbau des Display-Files und des Printerbuffers eine Rolle.
Um in einem Programm die 24-COPY durchzuführen, setzt man einfach die Zeilen 110 bis 140 hinter den eigentlichen COPY-Befehl.

```
      Zeile 0
      Zeile 1
      Zeile 2
      Zeile 3
      Zeile 4
      Zeile 5
      Zeile 6
      Zeile 7
      Zeile 8
      Zeile 9
      Zeile 10
      Zeile 11
      Zeile 12
      Zeile 13
      Zeile 14
      Zeile 15
      Zeile 16
      Zeile 17
      Zeile 18
      Zeile 19
      Zeile 20
      Zeile 21
Das ist die Zeile 22
      und das ist die Zeile 23
```

```
   1 REM ***********************
   5 REM *                     *
  10 REM *    24-Zeilen-COPY   *
  20 REM *                     *
  30 REM *                     *
  40 REM * ©1984 JOACHIM MILTZ *
  50 REM *                     *
  60 REM *      Hauptstr. 96   *
  70 REM *      5401  Wolken   *
  80 REM *                     *
  90 REM ***********************
  95 REM
 100 CLS : GO TO 300
 105 COPY
 110 FOR z=20672 TO 20704 STEP 3
2
 115 LET z1=z
 120 FOR i=0 TO 7
 125 FOR j=0 TO 31
 130 POKE 23296+j+i*32,PEEK (z1+
j)
 135 NEXT j: LET z1=z1+256
 140 NEXT i: LPRINT : NEXT z
 190 PAUSE 100: STOP
 300 REM    DEMONSTRATION
 310 FOR i=0 TO 21
 330 PRINT AT i,6;"Zeile ";i
 340 NEXT i
 350 PLOT 0,0: DRAW 255,0: DRAW
0,175. DRAW -255,0. DRAW 0,-175
 360 PRINT #0;AT 0,0;"Das ist di
e Zeile 22"
 365 PRINT #0;AT 1,6;"und das is
t die Zeile 23"
 370 GO TO 105
 900 SAVE "COPY 24" LINE 100
 910 VERIFY "COPY 24"
```

TI-99

Richtigstellung der in Computronic Heft 6 auf Seite 5 abgedruckten >Tips & Tricks< für den TI-99/4A.
Call Files

Beschreibung:
Das FILES-Unterprogramm verwendet man zur Änderung der Anzahl der externen Dateien, die gleichzeitig geöffnet sein können. Die Anzahl muß eine einstellige ganze Zahl (von 1 bis 9) sein. Der Befehl NEW muß im Direktbetrieb unmittelbar dem CALL FILES-Befehl folgen. Wird der CALL FILES-Befehl nicht eingegeben, so ist der Standardwert der Anzahl der möglichen offenen Dateien 3, sofern ein Peripheriegerät angeschlossen ist.
Das FILES-Unterprogramm ist vor allem im Zusammenhang mit dem Diskettenbetrieb interessant. Pro geöffneter Datei wird der für Programme verfügbare Speicherplatz im RAM-Bereich des Computers um 518 Bytes reduziert. (Bei zugeschaltetem Disk-Controller kann also der verfügbare Speicherplatz mit CALL FILES(1) oder CALL FILES(2) erhöht werden!)

Achtung!
Im TI-BASIC ist der CALL FILES-Befehl nur im Direktbetrieb (zusammen mit NEW) möglich. Die Anwendung des FILES-Unterprogramms in einem Programm kann unvorhersehbare Folgen haben – einschließlich Verlust eines Programmes und/oder Verlust von Daten auf der Diskette! CALL FILES (Anzahl) ist in den Peripheriegeräten implementiert. Es existiert nicht bei bloßem Anschluß eines Kassettenrecorders.

CALL FILES – Das FILES-Unterprogramm verwendet man zur Änderung der Anzahl der Dateien (max. 9), die gleichzeitig offen sein können. Der Standardwert der zulässigen offenen Dateien in TI BASIC liegt bei drei. Der für Programme verfügbare Speicherplatz im RAM wird um 534 Byte reduziert, plus 518 weitere Bytes für jede zulässige offene Datei.
Der CALL FILES Befehl hat folgende allgemeine Form:
CALL FILES (Anzahl)
NEW
Die Anzahl muß eine einstellige Zahl von 1 bis 9 sein. Sie gibt die Menge der Dateien an, die gleichzeitig offen sein können. Der Befehl NEW muß unmittelbar nach dem CALL FILES Befehl folgen. Wird CALL FILES nicht durchgeführt, liegt der Standardwert der möglichen offenen Dateien bei drei.

Vorsicht
CALL FILES kann nur als Befehl in TI BASIC verwendet werden. Es ist nicht zulässig, CALL FILES als Statement (Anweisung) in einem TI BASIC Programm einzugeben. Die Anwendung von CALL FILES in einem Programm ohne das unmittelbar folgende NEW kann unvorhergesehene Folgen haben, einschließlich Verlust eines Programms und/oder Verlust von Daten auf einer Diskette.

FORTH Kurs Teil 5

Bedingte Verzweigungen

Das Forth-Befehlswort „IF“ eröffnet alle bedingten Verzweigungen in FORTH-Programmen. Dieses Befehlswort erwartet eine 16-Bit Zahl auf dem Stack und behandelt dieses als Flag. Der „IF“-Befehl arbeitet nur zusammen mit dem Befehlswort „ENDIF“. „ELSE“ ist ebenfalls möglich, jedoch nicht unbedingt nötig. Folgendes Beispiel zeigt den Zusammenhnag der einzelnen Befehle:

```
IF      (IF testet das Flag; falles es
        wahr (= 1) ist ...)
...     (werden die nebenstehenden
        Anweisungen ausgeführt
...     (ansonsten)
ENDIF(fortgefahren)

IF
...     (falls Flag = 1, werden neben-
        stehende Worte)
...     (bis ELSE ausgeführt)
ELSE
...     (andernfalls erfolgt die Ausfüh-
        rung der Worte)
...     (zwischen ELSE und ENDIF)
ENDIF
```

Bei der Verwendung von Verzweigungen sollte man immer bedenken, daß diese nicht interprevativ verwendet werden dürfen, also nur innerhalb einer Definition. Probieren wir doch einmal die folgende, einfache Definition:

```
:DIVISION(Division mit Kommentar,
         falls Rest nicht Null) (Para-
         meter wie bei / – Befehl)
/MOD     (Division mit Rest)
SWAP     (Bringe Rest zum TOS und
         teste ihn als Flag)
IF."DIVISION geht nicht auf"
ENDIF
;        (falls Rest = „wahr“, wird
         der Text gedruckt)
```

Das nun definierte Wort „DIVISION“ erwartet also zwei Zahlen auf dem Stack, dividiert diese und beläßt den Quotienten auf dem TOS. Die Syntax des neuen Befehls entspricht genau dem Original "/" Befehl. Im Gegensatz zu diesem, wird jedoch bei Entstehung eines Restes die Meldung „DIVISION GEHT NICHT AUF“, ausgegeben.

Wir wollen nun ein Beispiel zeigen, welches den „ELSE“-Befehl vorführt. Es handelt sich hier um einen Vergleichs-Befehl, welcher angibt, ob die erste Zahl kleiner oder größer als die zweite Zahl ist.

```
:VERGLEICHE (Vergleich zweier
            Zahlen)
<           ."(FLAG ermitteln)
IF          ."KLEINER"
ELSE        ."GROESSER"
ENDIF
;
```

Beispiel:
33 100 VERGLEICHE (RETURN) KLEINER OK
100 33 VERGLEICHE (RETURN) GROESSER OK

Zum Vergleich dasselbe noch einmal in der Programmiersprache Basic:

```
10 INPUT „ZAHL1“;Z1

20 INPUT „ZAHL2“;Z2

30 IF Z1<Z2 THEN PRINT „KLEINER“

40 IF Z1>Z2 THEN PRINT „GROESSER“

50 END
```

Da im Standard-Basic keine „ELSE“-Befehl zugelassen ist, muß man sich mit zwei „IF“ Anweisungen behelfen, wenn man nicht zu dem UNSAUBEREN „GOTO“-Befehl greifen will. Dies geht selbstverständlich auf Kosten der Zeit.

Im übrigen sei an dieser Stelle darauf hingewiesen, daß einige FORTH-Versionen statt des Befehles „ENDIF“ den Befehl „THEN“ verwenden. Dies muß in dem entsprechenden Manual nachgelesen werden.

Wir haben nun einiges über die „IF“-Anweisung gelernt und wollen uns nun die eigentlichen Vergleichsbefehle näher ansehen:

< (n1 n2 n3 --> n1 f)
Flag wird 1 wenn die Zahl n2 kleiner als n3 ist.

> (n1 n2 n3 --> n1 f)
Flag wird 1 wenn die Zahl n2 größer als n3 ist.

= (n1 n2 n3 --> n1 f)
Flag wird 1 wenn beide Zahlen gleich sind

0< (n1 n2 n3 --> n1 n2 f)
Flag wird 1 wenn n3 eine negative Zahl ist

0= (n1 n2 n3 --> n1 n2 n3)
Flag wird 1 wenn die Zahl n3 gleich null ist.

Als Ergebnis des Vergleiches wird immer eine Boolsche Zahl abgelegt. Diese wird auch als Flag bezeichnet und gibt an, ob die Aussage richtig oder falsch ist. Ist diese richtig, so wird eine 1, ansonsten eine 0 auf den Stack gelegt.

Beispiele:

```
23 30 < . <RETURN> 1 OK
30 23 < . <RETURN> 0 OK
30 44 = . <RETURN> 0 OK
-2  0 < . <RETURN> 1 OK
```

Bei dem Vergleich zweier Zahlen sollte man bedenken, daß diese vom Vergleichsbefehl überschrieben werden. Will man diese jedoch für weitere Berechnungen erhalten, so kann man dies mit dem Befehl OVER erreichen:
23 30 OVER OVER < . . . <RETURN> 1 30 23

Programmschleife DO...LOOP

Ein wichtiger Bestandteil aller Programmiersprachen sind Schleifen. Unter Schleife versteht man einen oder mehrere Befehle, die durch zwei Kennzeichnungen eingeklammert sind. Diese eingeklammerten Befehle werden beliebig oft durchlaufen. Ist eine bestimmte Anzahl durchlaufen, wird das Programm fortgesetzt. In FORTH gibt es genau wie in Pascal keinen „GOTO“ Befehl, deshalb ist der Programmierer auf die Schleife angewiesen.

Der Schleifenbefehl hat in FORTH folgende Syntax:

DO ... LOOP (n1 n2 n3 --> n1)
Die Schleife wird von n3 bis n2 in Einerschritten durchlaufen. Die Zahl n3 muß in diesem Fall kleiner als n2 sein.

Die oberste Zahl auf dem Stack wird immer als Anfangswert und die darunterliegende Zahl als Endwert verwendet. Während der Schleife wird der Zahlenparameter immer um eins erhöht, deshalb muß der Endwert immer größer als der Anfangswert sein. Ist der Zahlenparameter größer oder gleich dem Endwert, so wird die Schleife verlassen. Da diese Kontrolle jedoch immer erst am Ende einer Schleife durchgeführt wird, ist die Schleife bereits einmal durchlaufen.
Will man während eines Schleifendurchlaufs den Zahlenparameter auslesen, kann man dies mit Hilfe des Wortes „I“. Dieses Wort legt den Zahlenparameter auf den Stack.

Wie der „IF“ Befehl kann auch die DO-LOOP-Schleife nicht interpretativ ausgeführt werden und muß deshalb innerhalb einer Definition stehen.
Das folgende Beispiel gibt die Zahlen 0–12 auf dem Schirm aus:

```
: ZAHLEN 12 0 DO I . LOOP ; <RETURN> OK

ZAHLEN <RETURN> 0 1 2 3 4 5 6 7
8 9 10 11 12 OK
```

Das gleiche in BASIC:
FOR I = 0 TO 12 : PRINT I, :NEXT 1

Ist der Endwert kleiner als der Anfangswert, so wird die Schleife nur einmal durchlaufen:

: ZAHLEN 0 12 DO I . LOOP ; <RETURN> OK
ZAHLEN <RETURN> 12 OK

Das nächste Beispiel zeigt zwei DO-LOOP-Schleifen die ineinander verschachtelt sind. Der Aufruf der neuen Definition gibt die Zahlen 0 bis 9 in einer Dreieckmatrix aus.

: SCHLEIFE1 0 DO I . LOOP ; <RETURN> OK

: SCHLEIFE2 0 DO CR 1 – DUP SCHLEIFE1 LOOP ; <RETURN> OK
11 10 SCHLEIFE2 <RETURN> OK

```
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8
0 1 2 3 4 5 6 7
0 1 2 3 4 5 6
0 1 2 3 4 5
0 1 2 3 4
0 1 2 3
0 1 2
0 1
0 OK
```

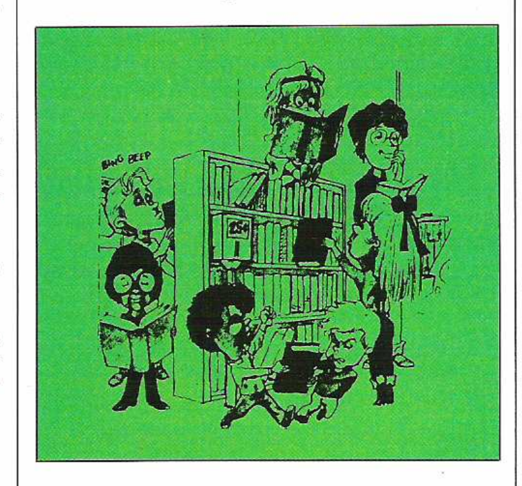
In vielen Programmen werden DO-LOOP-Schleifen zur Verzögerung von Programmabschnitten verwendet. Durch den folgenden Vergleich wird ersichtlich, wie schnell die Programmiersprache FORTH arbeitet:

Eine Leerschleife in FORTH:
: PAUSE 10000 0 DO LOOP ;

Die Ausführungszeit dieser Schleife beträgt im Durchschnitt 4 Sekunden. Je nach FORTH Version und Microprozessor kann diese Zeit etwas länger oder auch kürzer werden.

Die gleiche Schleife in BASIC:
10 FOR I = 0 TO 10000 : NEXT 1

In Basic ist die durchschnittliche Ausführungszeit etwa 4 bis 5 mal so lang wie in FORTH. Wie zu erkennen ist, eignet sich somit FORTH auch für Programme, in denen es auf genaue „Pause-Zeiten" ankommt. (Z. B. Eprom-Programmierung oder sonstige Einbzw. Ausgaben).
So mancher wird sich inzwischen fragen, ob man das Zahlenparameter nicht auch um 2, oder einer anderen Zahl erhöhen kann. Dies geschieht in Basic durch den Befehl „STEP".
Dies ist auch in FORTH möglich, indem man statt des „LOOP"-Befehls ein „+LOOP" einsetzt. In diesem Fall wird die Zahl, welche vor dem „+LOOP" auf dem Stack liegt, als Increment benutzt. Ist diese Zahl im negativen Bereich, so wird die Schleife bei jedem Durchlauf erniedrigt. In diesem Fall muß der Endwert kleiner als der Anfangswert sein, da sonst nur ein Durchlauf erfolgt.



Die Syntax: +LOOP (n1 n2 n3 --> n1 n2)

Einige Beispiele:

: ZAHLEN 10 0 DO I . 2 +LOOP ; <RETURN> OK

ZAHLEN <RETURN> 0 2 4 6 8 OK
In BASIC:

10 FORI = 0 TO 10 STEP 2 : PRINT I, : NEXT I

Umgekehrte Schleife:

: ZAHLEN 0 10 DO I . –2 +LOOP ; <RETURN> OK

ZAHLEN <RETURN> 10 8 6 4 2 OK

In Basic:

10 FORI = 10 TO 2 STEP –2 : PRINT I, : NEXT I

Bei Verwendung des „+LOOP" sollte man daran denken, daß vor dessen Ausführung eine 0 zur Endlosschleife führt.

BEGIN...UNTIL-Schleife

Da in FORTH kein Sprungbefehl vorhanden ist, müssen einige spezielle Schleifen diese ersetzen. Eine davon ist die BEGIN-UNTIL-Schleife.
Diese Schleife wird so lange durchlaufen, bis vor dem Befehl „UNTIL" ein „WAHR-Flag" gesetzt wird. Eine 0 führt zu einem neuen Durchlauf. Diese BEGIN-UNTIL-Struktur kann also sehr leicht den „GOTO"-Befehl ersetzen und führt somit zu einem übersichtlicheren Programm.
Die Struktur der Schleife sieht wie folgt aus:

```
BEGIN
  ...        (Befehle innerhalb der
              Schleife)
  ...

  ...
  ...        (Testteil der Schleife z. B.
              Vergleich)

             (Flag wird auf Stack gelegt)
UNTIL        (Ist Flag=0 erfolgt Sprung
              nach Begin)

  ...        (Flag ist ungleich 0; Pro-
              gramm wird fortgesetzt)
  ...
```

Das Beispiel mit den Zahlen:

```
: ZAHLEN
    0
    BEGIN
      1+ (Zahl um 1 erhöhen)
      DUP .      (Zahl ausgeben)
      DUP 10 =   (Ist 10 erreicht?)
    UNTIL
;
```

ZAHLEN <RETURN> 1 2 3 4 5 6 7 8 9 10 OK

In dem obigen Beispiel wird die Schleife so lange wiederholt, bis die auf dem Stack liegende Zahl den Wert 10 erreicht hat.

Eine weitere Anwendung wäre beispielsweise das Abfragen der Tastatur:

```
: INKEY (Auf <RETURN> wird gewartet)
    BEGIN
      KEY
      13 = (Ist <RETURN> gedrückt)
    UNTIL
  . "RETURN WURDE GEDRUECKT"
;
```

INKEY <RETURN>

<RETURN> RETURN WURDE GEDRUECKT OK

Zum Schluß das Beispiel noch einmal in Basic:

10 GET E$: IF E$="" THEN 10

20 IF ASC(E$(<> 13 THEN 10

30 PRINT „RETURN WURDE GEDRUECKT"

Erfreulich

Hiermit möchte ich Ihnen mitteilen, daß ich die Nachlieferung der Apple Disketten # 4 und # 5 erhalten habe. Weiterhin möchte ich mich bei Ihnen für den schnellen Service vielmals bedanken! Es ist heutzutage leider nicht mehr üblich auf Reklamationen so schnell zu reagieren und Abhilfe zu leisten. Deshalb VIELEN DANK!!! Holger Zeißler, Alsbach

Ärgerlich

Zu den Programmen „Schneevogel" Heft 6 und „Chopper-Flight" Heft 7, für Atari

Zu meinem großen Erstaunen habe ich festgestellt, daß in dem Listing „Schneevogel" für Atari-Computer in „Computronic" Nr. 6, ein Richard Käfer als Copyright-Inhaber angegeben ist. Dies entspricht jedoch nicht der Wahrheit!
Dieses Spiel ist schon im Januar '84 in einem amerikanischen Magazin erschienen. Denzufolge liegen die Urheberrechte bei dem dortigen Autor. Ich bitte Sie um Überprüfung dieser Tatsache! Jörg Detlefs, Hamburg

Ich möchte Sie hiermit darauf aufmerksam machen, daß das Listing „Schneevogel" nicht von Richard Käfer, sondern von John Weber, stammt. Das Listing ist vor längerer Zeit in einer amerikanischen Zeitschrift unter dem Namen „Snowbird" erschienen. Lars Kubitz, Berlin

Als Anlage eine Kopie des Originalprogrammes „Chopper-Flight" aus Amerika („Centurion"). Das Programm soll nämlich in Rot auf dem Bildschirm erscheinen. Das Bild ist aber nur schwarz/weiß. Poke 755, 224 ist falsch. Joachim Maiter

Redaktion:
Zunächst bedanken wir uns für die freundlichen Hinweise, wonach die Listings „Schneevogel" und „Chopper-Flight" für Atari, Heft 6+7, offensichtlich einen anderen Copyright-Inhaber haben als der von uns angegebene.
Unsere Vertragsbedingungen sehen vor, daß der Autor uns die Zusicherung gibt, daß er der alleinige Verfasser ist und dementsprechend über die weitere Verwendung eigenständig bestimmen kann.
Diese Zustimmung wurde uns erteilt. Wir können natürlich nicht jede auf dem Markt erscheinende Computer-Zeitschrift auf den Inhalt überprüfen. Insofern dürfen wir uns auf die zunächst gemachten Angaben verlassen. Allerdings wird unser Haus es nicht erlauben, daß mit solchen Praktiken wir und Sie, lieber Leser, getäuscht werden. Wir werden daher eine Prüfung durchführen, die unter Umständen zu einem Schadenersatz führen wird.

Ergänzung

Für die Veröffentlichung meines Mathematik-Programms möchte ich mich ganz herzlich bei Ihnen bedanken. Gleichzeitig bitte ich aber, den folgenden Nachtrag noch in einer Ihrer nächsten Ausgaben zu bringen! Grund dafür sind Bedienungsfehler Ihrerseits bei den Beispielen, die die Leistungsfähigkeit des Programms bei weitem nicht ausschöpfen. Vielleicht liegt es auch an der Knappheit der Dokumentation, die für „Hobbymathematiker" etwas unzureichend ist.

Nachtrag zum Mathematik-Programm – für ZX-Spektrum
1. Bei der Funktion $y = x^3 - 8x + x^2$ (S. 36) fällt auf, daß diese sich nur auf einen kleinen Teil der X-Achse verteilt. Grund dafür ist eine viel zu hohe Vergrößerung, so daß nur ein Teil der X – Y Paare abgebildet wird, und der letzte Pixel bei x = 2.5 gedruckt wird, obwohl der Definitionsbereich bis 15 gehen sollte. Hier fehlt also ein Teil der Funktion. Wird eine zu kleine Vertikaldehnung gewählt, so wird der GaPh nur etwas kleiner abgebildet, wie zum Beispiel die Sinusfunktion auf Seite 36. In allen Fällen stimmen aber die Koordination, angegeben mit A, B, P und Q!!
Das Problem der optimalen Vergrößerung könnte zwar softwaremäßig leicht gelöst werden, jedoch sehr auf Kosten der Geschwindigkeit. Außerdem ergäben sich große Probleme mit Unstetigkeitsstellen. Auf jeden Fall liegt hier kein Programmierfehler vor, was man bei einer neunten Version auch ausschließen sollte.
2. Auch wenn das Programm weitgehend immun gegen Unstetigkeitsstellen ist, so kann ein Auftauchen der Fehlermeldung „Number too big", oder ein Hinweis auf eine mögliche Definitionslücke nicht ausgeschlossen werden, denn manch eine unstetige Funktion kann selbst mit den raffiniertesten Mitteln der höheren Mathematik nicht gelöst werden! Und mein Programm kann nun einmal gewisse Gesetzmäßigkeiten der Mathematik nicht sprengen!
Abschließend möchte ich auf einen Fehler im Inhaltsverzeichnis hinweisen, wo die Lösbarkeit von Differentialgleichungen angepriesen wird. Richtig müßte es an dieser Stelle Differentialrechnung heißen im Sinne der ersten Ableitung.
Soweit die wichtige Ergänzung.
Uwe Haferland

Ein Tip:

Läßt sich mal eine Kassette nicht laden (Meldung: ERROR in DATA oder DATA NOT FOUND, beim TI-99/4A), dann sollte man versuchen die Kassette mit einem Stereogerät und zwar abwechselnd mit dem rechten oder linken Kanal zu laden – dies geht mit Hilfe des „Balance-Reglers" und eines entsprechenden Stereo-Kabels. Franz Sehi, Fürstenfeldbruck

Begeistert

Mit wachsender Begeisterung lese ich Ihr Computronic-Magazin und ich muß sagen „Sie haben eine Menge drauf" mit den abgedruckten Programmen.
Besonders gut fand ich das Programm für den Spectrum. Ich habe nun einige Änderungen am Listing durchgeführt, sodaß es mit dem Kempston-Joystick spielbar ist.
Meine Frage ist jetzt: Gibt es auch für solche Programmiertricks ein Honorar oder nur für ganze Programme?
Ich frage deshalb, weil ich selbst noch nicht in der Lage bin, Programme zu schreiben, sondern nur einige Änderungen oder Verbesserungen machen kann, die der besseren Spielbarkeit dienen. Harald Schüler, 6900 Heidelberg 1

Redaktion:
Diese Frage wird uns häufig gestellt. Natürlich bezahlen wir unseren Lesern auch gute Programmiertricks. Das Honorar dafür richtet sich nach Länge und Qualität des eingesandten Programmiertips.

Service

Meine TI-99-Kassetten habe ich prompt erhalten. Leider kann ich sie nicht lesen. Nach ca. 10 Std. Versuchen mit unterschiedlichen Recordern habe ich aufgegeben. In Abhängigkeit von der eingestellten Lautstärke erhalte ich entweder „NO DATA FOUND" oder „ERROR IN DATA DETECTED". Diese zweite Fehlermeldung kommt entweder gleich am Anfang bei dem etwas längeren Piepser oder erst nach einiger Zeit während des Einlesens. Vielleicht sollte ich noch darauf hinweisen, daß die eingelesenen Daten über den Monitorlautsprecher zu hören waren. Ich hoffe sehr, daß Sie mir helfen können. Dr. A. Ruff, 7400 Tübingen

Redaktion
Zwar sind wir weitgehendst bemüht, bespielte Cassetten bzw. Disketten im einwandfreien Zustand an unsere Kunden weiterzugeben, jedoch kann es auch bei uns vorkommen, daß ein Datenträger nicht einwandfrei versendet wird. Tatsache ist jedoch, daß jeder Datenträger auf seine Funktion überprüft wird. Durch eine Reihe anderer Umstände wie Tonkopfverschmutzung, unterschiedliche Tonkopfeinstellung, magn. Einflüsse etc. kann dieser Ladefehler Ihrer Kassetten jedoch zustande gekommen sein. Unser Angebot: Senden Sie die defekten Kassetten zurück, Sie erhalten umgehend kostenlos neue Exemplare zurück.





Ein Spiel in 378 Bildern

Vor langer Zeit lebte ein Volk, welches MAYA genannt wurde. Dieses Volk soll unermeßliche Schätze besessen haben.

Jetzt endlich hat sich jemand bereiterklärt, nach diesen sagenumwobenen Schätzen zu suchen. Dieser Jemand ist Coconut Joe. Für die abenteuerliche Expedition braucht er jedoch noch einen Partner. Wie wäre es zum Beispiel mit Ihnen? Begleiten Sie Coconut Joe bei seiner Schatzsuche durch den Urwald, und vielleicht haben Sie Glück und kommen in die Wüste zu den Pyramiden der MAYA, wo Sie weitere Schätze finden können.

Das Spiel läuft auf dem TI 99/4 A mit dem Extended BASIC Modul und wird mittels Joystick 1 gesteuert.

Was Sie jetzt lesen, werden Sie wohl kaum glauben wollen, es ist jedoch wahr:

Der Rechner kann in diesem Spiel bis zu 378!! verschiedene Bildschirmszenen darstellen. Ferner verfügt das Spiel über eine hervorragende grafische Darstellung.
Aus Speicherplatzgründen mußte jedoch auf eine Spielanleitung im Programm verzichtet werden, da das Programm den Speicher bis auf 654 Bytes ausnutzt.

Doch nun zum eigentlichen Spiel:
Tippen Sie das Listing ein oder laden Sie das Programm von Cassette. Danach starten Sie mit „RUN". Es dauert jetzt einige Sekunden, bis der Rechner die internen Vorbereitungen zum Spiel getroffen hat. Jetzt erscheint das Titelbild und El Condor Pasa wird gespielt. Um das Spiel zu beginnen, nehmen Sie Joystick 1 in die Hand und drücken den roten Aktionsknopf. Die Musik verstummt und nach wenigen Sekunden erscheint Coconut Joe am rechten Bildschirmrand. Nachfolgend werden nun die einzelnen Spielelemente beschrieben. Das ist die einfachste Art, das Spiel zu erklären.

Coconut Joe

Coconut Joe wird vom Spieler mittels des Joysticks gesteuert: Nach rechts und links. Wenn Sie im Bereich einer Leiter stehen, so bewegen Sie Joe mit dem Joystick nach oben oder unten, die Leiter hinauf oder hinunter.
Wenn Sie den Aktionsknopf drücken, so springt Joe in die Höhe. Gleichzeitiges Bewegen des Joysticks nach rechts oder links läßt Joe in die gewünschte Richtung springen.

Beachten Sie jedoch, daß Coconut Joe nur auf der Urwald- bzw. Wüstenoberfläche springen kann. In den unterirdischen Katakomben kann Joe nicht springen.

Schlange
Die Schlange kommt sowohl im Urwald als auch in der Wüste vor. Wenn Joe diese Schlange überspringt, so bekommen Sie 100 Punkte auf Ihrem Punktekonto gutgeschrieben. Ferner erhöht sich Ihr Überlebensbonus um 300 Punkte.
Berühren Sie jedoch die Schlange, ohne zu springen, so verliert Coconut Joe sein Leben und das Spiel ist zu Ende.

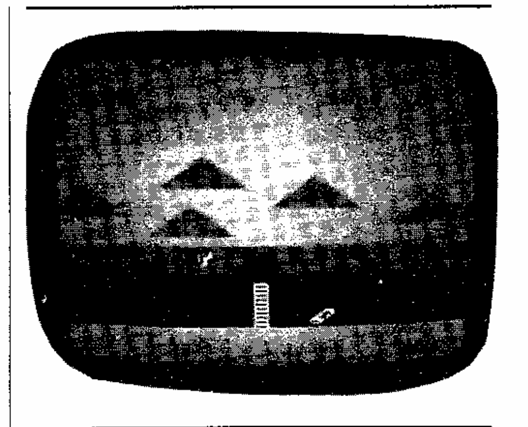
Rollender Baumstamm
Der rollende Baumstamm kommt nur im Urwald vor. Er rollt immer von links nach rechts. Wenn Joe den Baumstamm überspringt, so bekommt er 50 Punkte auf dem Punktekonto gutgeschrieben und 200 Überlebensbonuspunkte.
Wird Joe jedoch vom Baumstamm überrollt, so verliert er sein Leben und das Spiel ist zu Ende.

Urwaldfeuer
Das Urwaldfeuer kommt, wie der Name schon sagt, nur im Urwald vor. Wird es übersprungen, so werden 30 Punkte auf dem Punktekonto gutgeschrieben und der Überlebensbonus erhöht sich um 100 Punkte.
Wird das Feuer jedoch berührt, ohne daß Joe springt, so verliert er sein Leben und das Spiel ist zu Ende.

Kokosnuß
Die Kokosnuß kann unser Coconut Joe nur im Urwald finden. Diese Kokosnuß muß Joe aufnehmen, um die Chance zu bekommen, einen Schatz bergen zu können. Nimmt Joe die Kokosnuß auf, so erhält er 10 Punkte auf dem Punktekonto gutgeschrieben. Ferner erhöht sich die Chance, einen Schatz bergen zu können, um 1 Punkt. Der Überlebensbonus wird hierdurch nicht erhöht. Die Kokosnuß wird aufgenommen, indem sie berührt wird. Wenn Coconut Joe die Kokosnuß jedoch überspringt, so werden vom Überlebensbonus 500 Punkte abgezogen, was beim nächsten Szenenwechsel zum Tode führen kann, falls der Lebensbonus auf 0 gefallen ist.

Kaktus
Der Kaktus kommt nur in der Wüste vor. Wenn er außer bei einem Sprung berührt wird, so stirbt unser Joe den Heldentod.
Wird der Kaktus jedoch übersprungen, so werden dem Punktekonto 40 Punkte gutgeschrieben und der Überlebensbonus erhöht sich um 100 Punkte.

Wasserstelle
Die Wasserstelle findet unser Joe ebenfalls nur in der Wüste. Hierfür gelten die gleichen Bedingungen wie für die Kokosnuß, nur daß beim Überspringen 1000 Punkte vom Überlebensbonus abgezogen werden.



Goldbarren
Den Goldbarren findet unser Joe nur in den unterirdischen Katakomben. Wird der Goldbarren berührt, und die Chance einen Schatz bergen zu können ist größer 0, so verschwindet der Goldbarren und dem Punktekonto werden 5000 Punkte gutgeschrieben.
Wenn der Goldbarren jedoch berührt wird wenn die Chance 0 ist (d. h. wenn keine Kokosnuß oder Wasserstelle mehr zur Verfügung steht), stirbt Joe und das Spiel ist zu Ende. Wenn ein Goldbarren aufgenommen wird, so verringert sich die Chance um 1. Es kann also für jede angesammelte Kokosnuß oder Wasserstelle ein Schatz geborgen werden.

Silberbarren
Für den Silberbarren gelten die gleichen Bedingungen wie für den Goldbarren, nur gibt es beim Einsammeln 3000 Punkte auf dem Punktekonto.
Weiter gibt es im Spielgeschehen noch andere Hindernisse, so zum Beispiel die Fallgruben. Wenn Joe über eine Fallgrube läuft, ohne zu springen, so fällt er hinein und das Spiel ist zu Ende.
Wenn in einer Fallgrube eine Leiter steht, so kann Joe in die Katakomben gehen. Dazu muß der Joystick jedoch beim normalen Lauf schräg nach unten gezogen werden, da wenn Joe gerade über eine Fallgrube mit Leiter geht, er abstürzt und das Spiel zu Ende ist. In den unterirdischen Katakomben stehen auch Mauern, welche den Durchgang versperren. Wenn Joe eine Mauer berührt, so stirbt er, und das Spiel ist zu Ende. Coconut Joe hat schließlich nur ein einziges Leben.
Bei jedem Szenenwechsel wird der Überlebensbonus um 100 Punkte verringert. Wenn in diesem Moment der Bonus auf 0 steht, so ist das Spiel zu Ende. Ein Szenenwechsel verläuft immer dann, wenn Joe den rechten oder linken Bildschirmrand erreicht. Verläßt Joe den Bildschirm rechts, so erscheint er in der nächsten Szene links. Verläßt er den Bildschirm links, so erscheint er in der nächsten Szene rechts. Die Szenen werden durch den Zufall ausgewählt.



Die Anzeige
Die Anzeige ist am oberen Bildschirmrand eingeblendet. Links wird die Punktzahl angezeigt, in der Mitte ist die Chance zu sehen, welche anzeigt, ob und wieviele Schätze geborgen werden können.
Rechts ist der Überlebensbonus zu sehen. Bei Spielbeginn hat dieser Bonus den Wert von 2000 Punkten. Bei jedem Szenenwechsel verringert sich dieser Wert um 100 Punkte.
Die Punktzahl des Überlebensbonus kann nur in der oberen Ebene aufgefrischt werden. Seien Sie deshalb mit einem Szenenwechsel in den Katakomben vorsichtig, da man nie weiß, wann man wieder über eine Leiter ans Tageslicht gelangen kann. Man kann durch den Überlebensbonus in den Katakomben viel schneller das Leben verlieren als an der Oberfläche.
Das Spiel läuft zu Beginn nur im Urwald ab. Erst wenn Sie mindestens 50 000 Punkte erreicht haben, kommen Sie in die Wüste. Das ist kein einfaches Unterfangen. Das Spiel geht dann freilich nur noch in der Wüste weiter.
Doch jetzt genug der Erklärungen. Ich wünsche ein ERRORfreies Eintippen und viel Spaß beim Spiel. Nachfolgend noch einige Tips zum Eintippen:

1. Einige Zeilen sind länger als es der TI zuläßt. Tippen Sie die Zeile ein und schicken Sie diese mit <ENTER> ab. Danach holen Sie die Zeile mit <FCTN><REDO> wieder zurück und können den Rest eingeben.
2. Benutzer eines Diskettenlaufwerkes ohne Speichererweiterung sollten das Spiel bei abgeschalteter Peripheriebox eintippen und über einen Cassettenrecorder abspeichern, da durch den Dateibuffer des Laufwerkes der Speicher beim Programmlauf zu klein ist.
3. Wenn Sie ein Laufwerk mit Erweiterung besitzen, so ist vor dem Einlesen des Programmes von Cassette die Befehlsfolge: CALL FILES(1) <ENTER> NEW <ENTER> einzugeben, damit das Programm von Cassette eingelesen werden kann.
4. Während des Spielablaufes darf die <ALPHA LOCK> Taste nicht eingerastet sein, da sonst der Joystick nicht richtig abgefragt wird.
5. Tippen Sie vor allem die DATA Zeilen sehr genau ab, da ein hieraus resultierender Fehler sehr schwer zu finden ist.

Die genaue Programmstruktur ist der beigelegten Zuordnungstafel zu entnehmen. Nachfolgend wird aber zusätzlich noch eine Kurzbeschreibung des Programmes gegeben:

Zeile	Programmeinheit
1– 11	Copyright Zeilen
20– 133	Programmvorbereitung
135– 136	Aufruf des Titelbildes
137	Sprung ins eigentliche Spielprogramm
140– 170	Zufällige Auswahl einer der 378 verschiedenen Spielszenen
1000– 1110	Coconut Joe in der oberen Ebene
1500– 1520	Szenenwechsel in der oberen Ebene
2000– 2002	Kontrolle, ob Joe eine Leiter betritt
2005– 2010	Spielende durch Tod von Coconut Joe
3000– 3200	Kontrolle und Verzweigung, wenn Joe einen Gegenstand berührt
4000– 4030	Sprung von Coconut Joe
4500– 4530	Coconut Joe auf der Leiter
4550– 4640	Coconut Joe in der unteren Ebene
4800– 4820	Szenenwechsel in den Katakomben
4900	Schatz aufnehmen
4990– 4995	Betreten der Leiter in den Katakomben
5000– 5020	Sprung von der Leiter in die obere Ebene
6500– 6530	Abkontrolle des Titelbildes und Initialisierung der restlichen Grafik
10000–25030	Zusammenstellung der einzelnen Szenen
26000–26030	Punktanzeige
26040–26070	Anzeige der Chancen einen Schatz zu heben
26080–26110	Anzeige des Überlebensbonus
27000–27080	Ausgabe des Gonges bei Einsammeln eines Gegenstandes
28000–28014	Copyright
29900–29904	Abspeichern der Charakteren 48 bis 57
30000–30140	Bild ausgeben
30510–30575	Anzeige der Titelschrift
30800–30845	Generierung der Grafik Coconut Joe
31000–31160	Spielen des Titelliedes
31500–31540	Melodie vor Spielende
32000–32110	Belegung der Hauptgrafik

Wenn Coconut Joe gestorben ist, so erscheint wieder das Titelbild. Durch Druck auf den Aktionsknopf können Sie ein neues Spiel beginnen.
Ich hoffe, daß Sie genausoviel Spaß beim Spielen dieses abwechslungsreichen Spieles haben werden wie ich es habe und auch schon bei der Programmierung hatte.
ANMERKUNG: Pro Szene werden die oben genannten Punkte nur einmal vergeben, auch wenn der betreffende Gegenstand mehrmals übersprungen wird.
Wem es zu lange dauert, in die Wüste zu kommen, der kann die erforderliche Punktzahl in Zeile 142 entsprechend reduzieren.

```
1 REM  ***********************************************
2 REM  *                   M A Y A                   *
3 REM  *---------------------------------------------*
4 REM  * Ein Spiel in 378 Bildern fuer den TI 99/4 A *
5 REM  * geschrieben in Extended BASIC fuer Joysticks *
6 REM  *---------------------------------------------*
7 REM  * (c) 1984 by Volker Becker                   *
8 REM  *     Steinbacher Str.10,  6370 Oberursel 6   *
9 REM  ***********************************************
10 REM
11 REM
20 RANDOMIZE
40 CALL CLEAR :: CALL INITIAL(PAT$()):: CALL ASCIISET
110 CALL MAGNIFY(3):: CALL SCREEN(4):: FOR I=1 TO 14 :: CALL COLOR(I,1,1):: NEXT
 I
130 CALL BILD :: CALL CRIGHT(PAT$())
132 GOSUB 10000 :: GOSUB 11100 :: GOSUB 12000 :: CALL DELSPRITE(ALL)
133 P=0 :: M=0 :: Z=2100
135 CALL COLOR(3,2,13,4,2,13,12,11,7):: CALL TITEL
```

```
136 GOSUB 6500
137 GOTO 1000
140 CALL DELSPRITE(ALL)
141 SKP=0 :: Z=Z-100 :: CALL ZEIT(Z):: CALL PUNKTE(P):: CALL MOEGLICHKEIT(M):: I
F Z=0 THEN CALL KAMERAD :: GOTO 132
142 IF P<50000 THEN 145 ELSE GOSUB 10200 :: GOTO 150
145 ON INT(RND+1.5)GOSUB 10000,10100
150 ON INT(RND*2+1.5)GOSUB 11000,11100,11200
160 ON INT(RND*2+1.5)GOSUB 12000,12100,12200
165 IF INT(RND*19+1.5)=5 THEN CALL SPRITE(#3,36,9,176,85):: P2=5000 :: VAR=0 ELS
E IF INT(RND*19+1.5)=17 THEN CALL SPRITE(#3,36,15,176,157):: P2=3000 :: VAR=2
170 RETURN
1000 GOSUB 140
1010 CALL SPRITE(#1,68,13,122,230)
1020 CALL JOYST(1,X,Y):: IF X=0 THEN 1040
1030 CALL SOUND(-5,523,0):: IF X<0 THEN CALL PATTERN(#1,68)ELSE CALL PATTERN(#1,
84)
1040 CALL MOTION(#1,0,X*2)
1045 CALL KEY(1,T,S):: IF T=18 THEN 4000
1050 CALL POSITION(#1,X,Y):: IF Y<12 OR Y>239 THEN 1500
1060 CALL GCHAR(18,(Y+9)/8,BV):: IF BV=BVO THEN 2000
1070 CALL JOYST(1,X,Y):: IF X=0 THEN 1090
1080 CALL SOUND(-5,261,0):: IF X<0 THEN CALL PATTERN(#1,76)ELSE CALL PATTERN(#1,
92)
1090 CALL MOTION(#1,0,X*2)
1095 CALL KEY(1,T,S):: IF T=18 THEN 4000
1100 CALL COINC(ALL,ST):: IF ST THEN 3000
1110 GOTO 1020
1500 GOSUB 140
1510 IF Y>239 THEN CALL SPRITE(#1,84,13,122,12)ELSE CALL SPRITE(#1,68,13,122,230
)
1520 GOTO 1020
2000 CALL GCHAR(20,(Y+9)/8,BV):: IF BV>135 THEN CALL JOYST(1,X,Y):: GOTO 2002 EL
SE 2005
2002 IF Y=-4 THEN 4500
2005 CALL MOTION(#1,0,0):: CALL PATTERN(#1,132):: CALL MOTION(#1,10,0)
2010 CALL KAMERAD :: GOTO 132
3000 CALL MOTION(#1,0,0)
3005 ON SP2 GOTO 3100,3200
3100 GOTO 2005
3200 M=M+1 :: CALL MOEGLICHKEIT(M):: CALL DELSPRITE(#2):: P=P+10 :: CALL PUNKTE(
P):: CALL GONG(1):: GOTO 1020
4000 STV=0 :: CALL POSITION(#1,X,Y):: CALL LOCATE(#1,112,Y)
4010 FOR I=110 TO 190 STEP 10 :: CALL SOUND(-100,I,0,I+2,0,I+4,0):: CALL COINC(A
LL,ST):: IF ST THEN STV=1
4015 NEXT I
4020 CALL POSITION(#1,X,Y):: CALL LOCATE(#1,122,Y)
4025 IF STV=1 AND SKP=0 THEN P=P+P1 :: CALL PUNKTE(P):: Z=Z+21 :: CALL ZEIT(Z)::
 SKP=SKP+1
4030 GOTO 1020
4500 CALL PATTERN(#1,132):: CALL MOTION(#1,0,0):: CALL LOCATE(#1,138,121)
4510 CALL POSITION(#1,X1,Y):: IF X1>=173 THEN 4550 ELSE IF X1<=137 THEN 5000
4520 CALL JOYST(1,X,Y):: CALL MOTION(#1,-Y*1.5,0):: IF Y=0 THEN 4530 ELSE CALL S
OUND(-50,523-X1,0)
4530 GOTO 4510
4550 CALL JOYST(1,X1,Y):: IF X=0 THEN 4570
4560 CALL SOUND(-5,523,0):: IF X1<0 THEN CALL PATTERN(#1,68)ELSE CALL PATTERN(#1
,84)
4570 CALL MOTION(#1,0,X1*2)
4580 CALL POSITION(#1,X,Y):: IF Y<12 OR Y>239 THEN 4800
4590 CALL GCHAR(21,(Y+9)/8,BV):: IF BV=BVU THEN 2005 ELSE IF BV>135 THEN 4990
```



```
4600 CALL JOYST(1,X,Y):: IF X=0 THEN 4620
4610 CALL SOUND(-5,261,0):: IF X<0 THEN CALL PATTERN(#1,76)ELSE CALL PATTERN(#1,
92)
4620 CALL MOTION(#1,0,X*2)
4630 CALL COINC(ALL,ST):: IF ST THEN 4900
4640 GOTO 4550
4800 GOSUB 140
4810 IF Y>239 THEN CALL SPRITE(#1,84,13,176,12)ELSE CALL SPRITE(#1,68,13,176,230
)
4820 GOTO 4550
4900 IF M>0 THEN CALL DELSPRITE(#3):: P=P+P2 :: CALL PUNKTE(P):: M=M-1 :: CALL M
OEGLICHKEIT(M):: CALL GONG(VAR):: GOTO 4550 :: ELSE 2005
4990 IF BVO<>81 THEN 4600
4992 CALL JOYST(1,X,Y):: IF Y=4 THEN 4995 ELSE 4600
4995 CALL PATTERN(#1,132):: CALL MOTION(#1,0,0):: CALL LOCATE(#1,172,121):: GOTO
 4510
5000 CALL MOTION(#1,0,0):: CALL JOYST(1,X,Y):: IF Y=-4 THEN 4520 ELSE IF X=0 THE
N 5000 ELSE 5010
5010 CALL LOCATE(#1,122,121+4*X):: IF X<0 THEN CALL PATTERN(#1,68)ELSE CALL PATT
ERN(#1,84)
5020 GOTO 1020
6500 CALL CONDORPASA(B(),1)
6505 CALL KEY(1,T,S):: IF T<>18 THEN 6505
6510 CALL ASCII2 :: CALL HCHAR(2,1,58,32)
6520 FOR I=48 TO 57 :: CALL CHAR(I,PAT$(I-48)):: NEXT I
6530 RETURN
10000 CALL COLOR(3,2,13,4,2,13,1,1,1,2,13,1,5,7,1,6,1,1):: CALL SCREEN(4):: GOSU
B 20000 :: RETURN ! 2 Baumstaemme
10100 CALL COLOR(3,2,13,4,2,13,1,1,1,2,13,1,5,1,1,6,7,1):: CALL SCREEN(4):: GOSU
B 20000 :: RETURN ! 1 Baumstamm
10200 CALL COLOR(3,2,12,4,2,12,1,11,1,2,1,1,5,1,1,6,1,1):: CALL SCREEN(12):: GOS
UB 25000 :: RETURN ! Pyramiden
11000 CALL COLOR(9,7,2,10,2,2):: BVU=96 :: RETURN ! linke Mauer
11100 CALL COLOR(9,2,2,10,7,2):: BVU=104 :: RETURN ! rechte Mauer
11200 CALL COLOR(9,2,2,10,2,2):: BVU=0 :: RETURN ! keine Mauer
12000 CALL COLOR(14,8,2,7,1,2,8,7,2,11,7,7,13,11,11):: BVO=81 :: RETURN ! 1 Fall
grube
12100 CALL COLOR(14,2,2,7,11,11,8,7,7,11,7,2,13,2,2):: BVO=129 :: RETURN ! 2 Fal
lgruben
12200 CALL COLOR(14,2,2,7,11,11,8,7,7,11,7,7,13,11,11):: BVO=0 :: RETURN ! keine
 Fallgrube
20000 ON INT(RND*4+1.5)GOTO 20100,20200,20300,20400,20500
20100 RETURN
20200 CALL SPRITE(#2,116,7,122,157,0,-2):: SP2=1 :: P1=100 :: Z1=300 :: RETURN !
 Schlange
20300 CALL SPRITE(#2,100,7,122,20,0,10):: SP2=1 :: P1=50 :: Z1=200 :: RETURN !ro
llender Baumstamm
20400 CALL SPRITE(#2,60,12,122,85):: SP2=1 :: P1=30 :: Z1=100 :: RETURN ! Feuer
20500 CALL SPRITE(#2,124,7,122,157):: SP2=2 :: P1=0 :: Z1=-500 :: RETURN ! Kokos
nuss
25000 ON INT(RND*3+1.5)GOTO 20100,25020,20200,25030
25020 CALL SPRITE(#2,108,5,122,85):: SP2=1 :: P1=40 :: Z1=100 :: RETURN ! Kaktus
25030 CALL SPRITE(#2,140,8,122,157):: SP2=2 :: P1=0 :: Z1=-1000 :: RETURN ! Wass
erstelle
26000 SUB PUNKTE(P)
26010 A$=STR$(P)
26020 DISPLAY AT(1,1)SIZE(LEN(A$)+3):A$&":::"
26030 SUBEND
26040 SUB MOEGLICHKEIT(M)
26045 IF M>99 THEN M=99
26050 A$=STR$(M)
```

```
26060 DISPLAY AT(1,14)SIZE(LEN(A$)+3):A$&":::"
26070 SUBEND
26080 SUB ZEIT(Z)
26085 IF Z<=0 THEN Z=0
26090 A$=STR$(Z)
26100 DISPLAY AT(1,20)SIZE(LEN(A$)+4):A$&"::::"
26110 SUBEND
27000 SUB GONG(VAR)
27005 CALL MOTION(#1,0,0)
27010 FOR I=0 TO 9 :: CALL SOUND(-250,391,I):: NEXT I
27020 IF VAR=1 THEN SUBEXIT
27030 FOR I=0 TO 9 :: CALL SOUND(-250,329,I,391,I+11):: NEXT I
27040 FOR I=0 TO 9 :: CALL SOUND(-250,261,I,329,I+11,391,I+21):: NEXT I
27050 IF VAR=2 THEN SUBEXIT
27060 FOR I=0 TO 9 :: CALL SOUND(-250,391,I,261,I+11,329,I+21):: NEXT I
27070 FOR I=0 TO 9 :: CALL SOUND(-250,261,I,391,I+11,329,I+21):: NEXT I
27080 SUBEND
28000 SUB CRIGHT(PAT$())
28002 DATA 48,3C4299A1A199423C,50,66A5242442432418,51,78A424382423A27C,53,00001C
24F821120C,54,00003C40C041423C,55,08141838D4151916
28004 DATA 57,0000334C88080808
28006 RESTORE 28002
28008 FOR I=1 TO 7 :: READ C,C$ :: CALL CHAR(C,C$):: NEXT I
28009 CALL CHAR(56,PAT$(8),52,PAT$(4))
28010 DATA 48,58,56,52,58,50,58,51,53,54,55,53,57
28012 FOR I=1 TO 13 :: READ C :: CALL HCHAR(2,I+9,C):: NEXT I
28014 SUBEND
29900 SUB INITIAL(PAT$())
29902 FOR I=48 TO 57 :: CALL CHARPAT(I,PAT$(I-48)):: NEXT I
29904 SUBEND
30000 SUB BILD
30010 CALL HCHAR(1,1,58,64):: CALL HCHAR(20,1,59,160)
30020 DISPLAY AT(3,1):"(........)(.......)(.........) (.......)  (....)  (.......)
  (..)      (..)      (..)        ,-          ,-          ,-"
30030 CALL VCHAR(7,7,65,9):: CALL VCHAR(7,8,66,9):: CALL VCHAR(7,16,73,9):: CALL
 VCHAR(7,17,74,9)
30040 CALL VCHAR(7,25,65,9):: CALL VCHAR(7,26,66,9)
30050 CALL HCHAR(16,1,120,96):: CALL HCHAR(19,1,121,32)
30060 DATA 7,129,112,25,129,112,16,81,89
30070 RESTORE 30060
30080 FOR I=1 TO 3 :: READ S,C1,C2
30090 CALL HCHAR(17,S,C1,2):: CALL HCHAR(18,S,C1,2):: CALL HCHAR(19,S,C2,2):: NE
XT I
30100 CALL VCHAR(20,5,96,5):: CALL VCHAR(20,28,104,5):: CALL VCHAR(20,16,136,5):
: CALL VCHAR(20,17,137,5)
30110 DATA 7,12,9,20,10,3,11,29,12,11
30120 FOR I=1 TO 5 :: READ Z,S :: CALL HCHAR(Z,S,33):: CALL HCHAR(Z,S+1,34):: CA
LL HCHAR(Z+1,S-1,33):: CALL HCHAR(Z+1,S,35,2):: CALL HCHAR(Z+1,S+2,34)
30130 CALL HCHAR(Z+2,S-2,33):: CALL HCHAR(Z+2,S-1,35,4):: CALL HCHAR(Z+2,S+3,34)
:: NEXT I
30140 SUBEND
30510 SUB TITEL
30520 CALL MAGNIFY(4)
30525 DATA ,0103060C1830600C,0103070F1B3363C3,83,0103070F1B3363C3,83030303030303030
3,0F0F18303030303,303F3F303030303,E1E1311919191818
30530 DATA 18F8F81818181818,8C8C8C8C8C8CD8D8,7070707070707007,80C0E0F0D8CCC6C3,C1
C0FFFFC0C0C0C,,80C0E0F0180C0603
30532 RESTORE 30525
30535 FOR I=67 TO 91 STEP 8 :: FOR K=1 TO 4
30540 READ A$ :: CALL CHAR(I+K,A$)
30545 NEXT K :: NEXT I
```




```
30550 J=64
30555 DATA 68,76,84,92
30560 FOR I=1 TO 4
30565 READ A :: CALL SPRITE(#I,A,5,89,J):: J=J+32
30570 NEXT I
30575 SUBEND
30800 SUB ASCII2
30805 CALL DELSPRITE(ALL)
30810 CALL MAGNIFY(3)
30815 DATA 68,000100000C0201,69,01030303010200C,70,E0E0A0E0202060E,71,E0E0E0E0404
040C,76,0001000000000708,77,0103030301010103
30820 DATA 78,E0E0A0E02020E0E,79,E0E0E0E04020101,84,0707050704040607,85,07070707
02020203,86,0080000030408
30825 DATA 87,80C0C0C080403,92,0707050704040707,93,0707070702040808,94,00800000
000F01,95,80C0C0C0808080C
30829 RESTORE 30815
30830 FOR I=1 TO 16
30835 READ C,C$ :: CALL CHAR(C,C$)
30840 NEXT I
30845 SUBEND
31000 SUB CONDORPASA(B(),KY)
31010 DATA 2,30000,30000,30000,30000,1,15000,293,1,15000,277,1,15000,293,1,15000
,329,1,15000,349,1,15000,329
31020 DATA 1,15000,349,1,15000,391,6,220,261,349,440,6,349,440,2,440,523,6,349,4
40,1,466,587,1,440,523
31030 DATA 6,349,440,1,15000,391,1,15000,349,4,174,220,293,349
31040 DATA 6,15000,293,2,15000,349,7,15000,293,1,15000,220
31041 DATA 2,30000,30000,30000,30000,1,15000,293,1,15000,277,1,15000,293,1,15000
,329,1,15000,349,1,15000,329
31042 DATA 1,15000,349,1,15000,391,6,220,261,349,440,6,349,440,2,440,523,6,349,4
40,1,466,587,1,440,523
31043 DATA 6,349,440,1,15000,391,1,15000,349,4,174,220,293,349
31044 DATA 6,15000,293,2,15000,349,7,15000,293,1,15000,440,4,15000,15000,15000,1
5000,3,466,587,1,440,554
31045 DATA 1,466,587,1,440,554,1,466,587,1,440,554,6,466,587,1,587,698,1,466,587
,4,220,261,349,440
31046 DATA 6,440,523,1,466,587,1,440,523,6,349,440,1,15000,391,1,15000,349,4,174
,220,293,349,6,15000,293
31047 DATA 2,15000,349,8,15000,293,0
31060 RESTORE 31010
31070 AZ,STV=0
31075 FOR I=0 TO 2 :: CALL SOUND(-600,220,I):: NEXT I
31080 READ K :: IF K=0 THEN 31060
31090 READ B(1),B(2),B(3),B(4)
31100 FOR I=1 TO K
31110 FOR J=1 TO 4
31120 STV=STV+1 :: IF STV>AZ THEN READ AZ,M1,M2 :: STV=1
31130 FOR L=0 TO 2 :: CALL SOUND(-600,M1,L+1,M2,L,B(J),9):: NEXT L :: CALL KEY(1
,T,S):: IF S<>0 THEN SUBEXIT
31140 NEXT J :: NEXT I
31150 GOTO 31080
31160 SUBEND
31500 SUB KAMERAD
31510 DATA 500,130,500,174,250,220,25,30000,100,220,25,30000,500,220,500,195,100
0,174,500,130
31520 RESTORE 31510
31525 CALL DELSPRITE(#1)
31530 FOR I=1 TO 10 :: READ L,T :: CALL SOUND(L,T,0):: NEXT I
31540 SUBEND
32000 SUB ASCIISET
32010 DATA ,0103070F1F3F7FFF,80C0E0F0F8FCFEFF,FFFFFFFFFFFFFFFF,000000103070F1D,
3E7DFFD7ABFE,007FFFD7AF96BC38,B060C08,FFFF7F7F3F1F0701
```



```
32020 DATA FFFFFFFEFCF8E08,030F1F3F3F7F7FFF,C0F0F8FCFCFEFEFF,FFFF7F7F3F3F1F1F,FF
FFFEFEFCFCF8F8,FFFFFFFFFFFFFFFF,,,,,,,,,,,,,FFFFFFFFFFFFFFFF
32030 DATA 0000416356 7D2B1E,0F07030003060C18,0002068ADE74B8F,E0C0800080C0603,,0F
0F1F1F0F0F0F0F,F0F0F0F0F8F8F8F,,,,,,,0F0F1F1F0F0F0F0F
32040 DATA F0F0F0F0F8F8F8F0,,,,,,,,,,,,,,,,FFFF,,,,,,,FBFBFB00DFDFDF,,,,00000000
103070A,153F4792BB924438,1C2E5FFBD7AEFCF8,F0A0408
32050 DATA FBFBFB00DFDFDF,,,,0303030303181B1B,1B1B1F0F03030303,00000C0C6C6C6C7C,
7860606060E0C,FFFF,,,,000E1B2E4,050A10150A000A05
32060 DATA 0000806030081008,50A00050A010A040,FFFFFFFFFFFFFFFF,,,,000000000001030
6,070D0D0706030301,000000000080C060,E0B0B0E060C0C08
32070 DATA FFFFFFFFFFFFFFFF,,,,0003070363110907,03070F1F02040818,0080C0808C1020C
,80C0F0F08040203,1F1F18181F1F1818,F8F81818F8F81818
32080 DATA ,,000000000030F3FFF,FF3F0F03,0000000C0F0FCFF,FFFCF0C
32090 RESTORE 32010
32100 FOR I=32 TO 143 :: READ C$ :: CALL CHAR(I,C$):: NEXT I
32110 SUBEND
```

ASC II – DEF

Definition von Charakteren, Teil 2
Umwandeln eines Codefeldes

Wenn Sie eine Datei mittels des Programmes ASCIIDEF oder des Unterprogrammes ACIIPAT erstellt haben, so muß diese Datei in einen für den Drucker lesbaren Grafikcode umgesetzt werden. Das ist so zu verstehen, daß für den Drucker ein zweiter Zeichensatz generiert wird. Dazu wird das Programm ASCIIUMW verwendet. Das Programm arbeitet nach folgendem Prinzip: Der im Codefeld enthaltene Hexadezimale Code wird in einen Dezimalen Code umgesetzt. Mittels dieses Dezimalcodes wird intern eine binäre 8 x 8 Punkte Matrix erstellt. Diese Matrix wird dann von Spalte 8 bis Spalte 1 von Zeile 1 bis 8 abgerastert und von jeder Spalte der dazugehörige Dezimalcode ermittelt. Die dem jeweiligen Dezimalcode zugeordneten Zeichen werden jetzt in einer Stringverknüpfung zusammengesetzt. So entsteht für jedes Zeichen intern ein aus 8 Zeichen bestehender String. Dieser String wird auf der Datei zusätzlich zu dem Hexadezimalcode abgespeichert und entspricht im Grafikmodus des Druckers exakt dem auf dem Bildschirm zugeordneten Zeichen.

Laden Sie das Programm ASCIIUMW ein und starten Sie dieses mit RUN. Sie werden nun aufgefordert den Namen des entsprechenden Codefeldes (z. B. DSK1.FELD oder CS1) einzutippen. Alles weitere erledigt der Rechner für Sie. Das umgewandelte Codefeld wird wieder unter dem gleichen Namen auf den entsprechenden Datenträger zurückgeschrieben. Auch ein umgewandeltes Codefeld kann mittels des Programmes ASCIIDEF eingelesen und editiert werden, muß jedoch wenn es zurückgeschrieben wird, erneut umgewandelt werden. Die Umwandlung eines Codefeldes muß in der Regel nur einmal erfolgen um den Zeichensatz für den Drucker zu belegen. Die Umwandlung dauert mittels Diskette etwa 25 Minuten, mit Recorder dagegen ca. 1 Stunde. Der Rechner informiert jedoch laufend, welches Zeichen im Moment umgewandelt wird. Die so umgewandelte Datei ist Voraussetzung für den Ausdruck des Zeichensatzes sowie für die Erstellung einer Hardcopy. Auch mit einer umgewandelten Datei kann mittels des Unterprogrammes ASCIISET der Zeichensatz umbelegt werden, da der Hexadezimalcode nicht zerstört wird.

Ausdruck des grafischen Zeichensatzes

Um den Zeichensatz auszudrucken, wird das Programm ASCIIPRINT herangezogen. Laden Sie das Programm ASCIIPRINT ein und starten Sie dieses mit RUN. Sie werden aufgefordert den Namen des Codefeldes einzugeben (z. B. DSK1.FELD oder CS1). Danach werden Sie nach dem Programmnamen gefragt, der in der Seitenüberschrift erscheinen soll. Nach Eingabe dieser Informationen erfolgt der Ausdruck. Achten Sie jedoch darauf, daß der Drucker vor dem Programmstart am Anfang einer neuen Seite steht, da das Programm den Seitenvorschub von 72 Zeilen beim Drucker festlegt, kurz nachdem es gestartet wurde.

Erstellen einer Hardcopy

Zuvor müssen einige Einschränkungen gemacht werden. Das Programm erstellt eine Hardcopy des Bildschirmes mit einem selbst belegten grafischen Zeichensatz. Dazu muß jedoch ein umgewandeltes Codefeld vorhanden sein. Eine Hardcopy wird nur von der auf dem Bildschirm erscheinenden Grafik erstellt, nicht jedoch von Sprites. Die Spritecharakteren müssen also, wenn sie nicht copiert werden sollen, mit CALL HCHAR oder CALL VCHAR vorher als normale Grafik auf den Bildschirm gebracht werden. Das Programm copiert den Bildschirm von Zeile 1 bis 24 und von Spalte 2 bis 31. Spalte 1 und 32 werden nicht copiert, sind jedoch auf dem Bildschirm meist auch nicht sichtbar, da sie die linke und rechte Begrenzung darstellen.

Doch nun zum eigentlichen Copiervorgang: Tippen Sie das Unterprogramm COPY zusätzlich zu Ihrem normalen Programm ein oder laden Sie dieses mit MERGE auf ein bestehendes Programm auf. Rufen Sie das Unterprogramm an gegebener Stelle im Programm auf mit Angabe der entsprechenden Datei, auf welche der Bildschirm gespeichert werden soll. So z. B. CALL COPY („DSK1.BILD") oder CALL COPY („CS1"). Bei der Verwendung eines Recorders ist jedoch Vorsicht geboten, da der Rechner die Steueranweisung auf dem Bildschirm ausgibt, und somit das zu copierende Bild scrollt. Es ist deshalb bei Recorderbetrieb besser den OPEN Befehl aus dem Unterprogramm herauszunehmen und OPEN #88:„CS1" ... in das Hauptprogramm vor die Generierung des Bildes zu schreiben. Das Unterprogramm heißt dann nur noch COPY und wird mit CALL COPY aufgerufen. Die so gespeicherten Copien können nur mit dem Programm COPYPRINT umgewandelt werden bzw. ausgedruckt werden.

Laden Sie zu diesem Zweck das Programm COPYPRINT und starten Sie dieses mit RUN. Sie erhalten dann ein Menü worin Sie auswählen können ob Sie eine Copie umwandeln oder ausdrucken wollen. Das Ausdrucken ist nur mit einem Diskettenlaufwerk möglich, da ein Recorder durch die Struktur des Rechners nur eine Satzlänge von 192 Bytes verarbeiten kann, jedoch eine umgewandelte Hardcopy 24 Sätze zu je 240 Bytes hat. Der Ausdruck einer Hardcopy erfolgt jedoch auch bei der Umwandlung. Da die eigentliche Umwandlung nur ca. 1½ Minuten dauert, können Benutzer eines Recorders über die Umwandlung die Copien ausdrucken. Wenn Sie die Umwandlung gewählt haben, so geben Sie die Datei ein, in welcher der Ze…



chensatz für den Drucker gespeichert ist. Der Rechner liest nun die Datei ein. Danach geben Sie die Datei ein, unter welcher Sie den Bildschirm abgespeichert haben. Der Rechner wandelt jetzt die Bildschirmcopie in ein auszudruckendes Bild um. Dazu wird der 8stellige Druckerstring für das entsprechende Zeichen gelesen, und diese Strings zusammengesetzt. Eine Bildschirmzeile besteht somit aus einem 240 Zeichen langen String. Nach der Umwandlung können Sie, sofern Sie ein Laufwerk besitzen, diese umgewandelte Copie abspeichern. Die Copie wird jedoch ausgedruckt, egal ob Sie abspeichern oder nicht. Wenn Sie weitere Umwandlungen mit dem gleichen Zeichensatz wünschen, so antworten Sie bei der Frage mit ‚J'. Dann geht es weiter mit dem Einlesen der Bildschirmcopie.
Dieses Programmpaket soll einige Anregungen bringen, was mit dem TI in EX-BASIC alles machbar ist. Einige Programme können ohne weiteres zusammengefaßt oder umgeändert werden, so daß z. B. eine Hardcopy direkt im Programm ausgedruckt werden könnte. Dies würde jedoch bei eigenen größeren Programmen eine Speichererweiterung voraussetzen. Diskettenlaufwerkbesitzer können auch anstatt der Sequentiellen Dateien Direktzugriffsdateien verwenden. Es wurden jedoch Sequentielle Dateien benutzt, da nur diese mit einem Recorder benutzt werden können, und das Programmpaket so vielen Benutzern wie möglich von Nutzen sei soll. Programmintern werden die Dateien sowieso als Direktzugriffsdateien benutzt. Dazu wird schließlich die komplette Datei in Felder eingelesen.
Ich würde mich freuen, wenn ich Ihnen die mühselige Arbeit der Zeichendefinition mit diesen Programmen erleichtern könnte.

```
10 REM ******************************
12 REM *       A S C I I U M W      *
14 REM * Umwandeln eines Codefeldes *
16 REM *----------------------------*
18 REM *    TI 99/4 A               *
20 REM *  + TI Extended BASIC       *
22 REM *  + externer Speicher       *
24 REM *----------------------------*
26 REM * (c) 1984 by Volker Becker  *
28 REM *      Steinbacher Str. 10   *
30 REM *      6370 Oberursel 6      *
32 REM ******************************
34 REM
36 REM
100 DIM ASCII$(112),G$(112)
110 DISPLAY AT(1,1)ERASE ALL:"UMWANDLUNG EINES CODEFELDES        ZUR GRAPHISCHEN
        DRUCKER-AUSGABE      ----------------------------"
120 DISPLAY AT(6,1):"NAME DES UMZUWANDELNDEN      CODEFELDES (GERAET.NAME)"
130 ACCEPT AT(8,1)BEEP:DATEI$
140 ON ERROR 130
150 OPEN #1:DATEI$,INPUT ,SEQUENTIAL,INTERNAL,FIXED 80
160 ON ERROR STOP
170 FOR I=32 TO 143
180 INPUT #1:G$(I-32),ASCII$(I-32)
190 NEXT I
200 CLOSE #1
210 OPEN #1:DATEI$,OUTPUT,SEQUENTIAL,INTERNAL,FIXED 80
220 DISPLAY AT(10,1):"UMWANDLUNG DES ZEICHENS->"
230 FOR K=32 TO 143
240 DISPLAY AT(10,26):USING "###":K
250 HEX$=ASCII$(K-32)
260 M=1
270 C$=""
280 H$=SEG$(HEX$,1,2)
290 E=0
300 FOR L=LEN(H$)TO 1 STEP -1
310 B$=SEG$(H$,L,1)
320 FOR J=0 TO 5
330 IF ASC(B$)=65+J THEN A=10+J :: GOTO 360
340 NEXT J
350 A=VAL(B$)
360 D=D+A*16^E
370 E=E+1
380 NEXT L
390 IF D=0 THEN FELD$(M)="00000000" :: GOTO 450
400 D=D/2
410 IF D=INT(D)THEN C$="0"&C$ :: GOTO 430
420 C$="1"&C$ :: D=INT(D)
```

```
430 IF INT(D)>0 THEN 400
440 FELD$(M)=C$ :: C$=""
450 M=M+1
460 HEX$=SEG$(HEX$,3,LEN(HEX$)-2)
470 IF HEX$<>"" THEN 280
480 D=0
490 FOR R=1 TO 8
500 IF LEN(FELD$(R))<8 THEN FELD$(R)="0"&FELD$(R):: GOTO 500
510 NEXT R
520 GRAPH$=""
530 FOR N=1 TO 8
540 FOR O=8 TO 1 STEP -1
550 B$=SEG$(FELD$(O),N,1)
560 D=D+VAL(B$)*2^(O-1)
570 NEXT O
580 GRAPH$=GRAPH$&CHR$(D)
590 D=0
600 NEXT N
610 G$(K-32)=GRAPH$
620 NEXT K
630 FOR I=32 TO 143
640 PRINT #1:G$(I-32),ASCII$(I-32)
650 NEXT I
660 CLOSE #1
670 DISPLAY AT(10,1):"NOCH EINE UMWANDLUNG (J/N)"
680 ACCEPT AT(10,28)VALIDATE("JN")SIZE(1)BEEP:SKP$
690 IF SKP$="J" THEN 110
700 CALL CLEAR
710 STOP
720 END
```

```
100 REM ***********************************************
110 REM *                A S C I I P R I N T          *
120 REM * Ausdrucken eines Codefeldes ( ASCII - Satz ) *
130 REM *---------------------------------------------*
140 REM *   T I  99/4  A                              *
150 REM * + TI Extended BASIC                         *
160 REM * + externer Speicher (Laufwerk oder Recorder) *
170 REM * + Seikosha GP 700 A (oder anderen kompartiblen*
180 REM *   Drucker mit 8 DOT Matrix                  *
190 REM *---------------------------------------------*
200 REM * (c) 1984 by Volker Becker                   *
210 REM *     Steinbacher Str. 10                     *
220 REM *     6370 Oberursel 6                        *
230 REM ***********************************************
240 REM
250 REM
260 REM Druckeroptionen
270 OPEN #99:"PIO" ! Eroeffnen der Paralellen Schnittstelle
280 N$=CHR$(27)&"A" ! Einstellen auf normalen Zeilenabstand
290 SL$=CHR$(27)&"ZH" ! Seitenlaenge auf 72 Zeilen einstellen
300 T10$=CHR$(16)&"010" ! Tabulator auf Position 10 setzen
310 T11$=CHR$(16)&"011" ! Tabulator auf Position 11 setzen
320 ZV$=CHR$(10)! Zeilenvorschub
330 SV$=CHR$(12)! Seitenvorschub
340 CR$=CHR$(13)! Wagenruecklauf ohne Zeilenvorschub
350 U$=CHR$(27)&"R"&CHR$(32)&"_" ! Unterstreichen mit dem Zeichen '_' (32 mal)
360 G$=CHR$(27)&"K008" ! Drucker auf 8 Graphikzeichen vorbereiten
370 REM Ende der Druckeroptionen
380 DISPLAY AT(1,1)ERASE ALL:" Ausdruck des ASCII Satzes    ------------------------
----"
390 DISPLAY AT(4,1):"ASCII Eingabedatei :         (Geraet.Dateiname)"
```



```
400 ACCEPT AT(6,1)BEEP:DATEI$
410 ON ERROR 400
420 OPEN #1:DATEI$,INPUT ,SEQUENTIAL,INTERNAL,FIXED 80
430 DISPLAY AT(8,1):"Programmname  :" :: ACCEPT AT(8,16)BEEP:PNAME$
440 PRINT #99:N$;SL$
450 PAGE=1 :: GOSUB 540
460 FOR I=32 TO 143 :: INPUT #1:GRAPH$,HEX$
470 PRINT #99:T11$;
480 PRINT #99,USING "###":I;
490 PRINT #99:"   ";HEX$;"      ";G$;GRAPH$
500 IF I-32=56 THEN PAGE=2 :: PRINT #99:SV$ :: GOSUB 540
510 NEXT I
520 CLOSE #1 :: CLOSE #99
530 STOP
540 PRINT #99:T10$;"Zeichensatz des.Programmes ";PNAME$;"   Seite :";PAGE;ZV$;ZV
$
550 PRINT #99:T10$;"ASCII      Hexadezimal     Zeichen";CR$;T10$;U$
560 RETURN
570 END
```

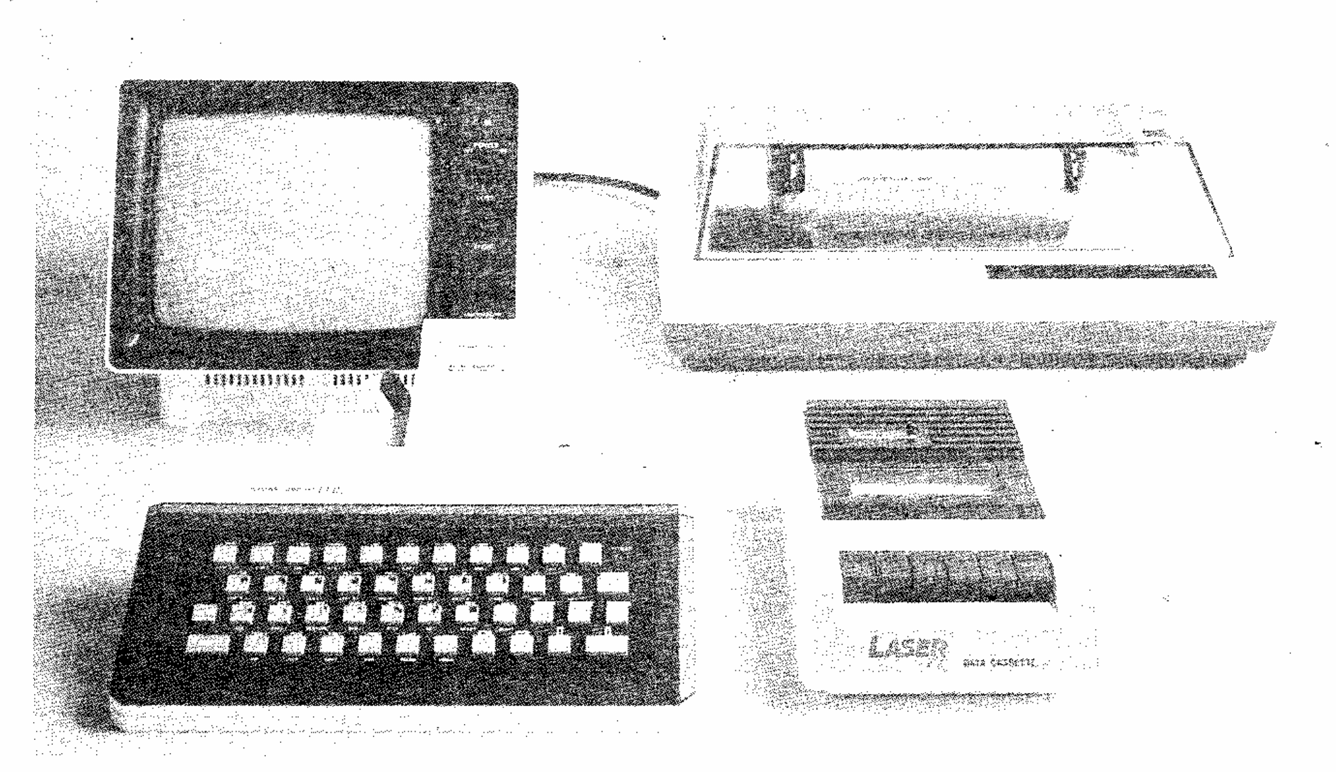
32100 SUB COPY (DATEI)

```
32110 !
32120 ! Unterprogramm zur Abspeicherung des Bildschirmes
32130 ! fuer TI 99/4 A
32140 !
32150 ! (c) 1984 by Volker Becker
32160 !
32170 OPEN #88:DATEI$,OUTPUT,SEQUENTIAL,INTERNAL,FIXED 80
32180 ! VORSICHT - Bei Verwendung eines Cassetenrtrecorders
32190 !            darf der OPEN Befehl nicht im Unterprogramm stehen,
32200 !            sondern muss im Hauptprogramm vor der generierung
32210 !            des Bildschirmes verwendet werden.
32220 !            Die erste Zeile des Unterprogrammes muss dann  heissen SUB CO
PY
32230 !            und es wird nur mit CALL COPY aufgerufen.
32240 !
32250 FOR I=1 TO 24
32255 GRAPH$=""
32260 FOR K=2 TO 31
32270 CALL GCHAR(I,K,CH)
32280 GRAPH$=GRAPH$&CHR$(CH)
32290 NEXT K
32300 PRINT #88:GRAPH$
32310 NEXT I
32315 CLOSE #88
32320 SUBEND
```

```
10 REM ***********************************************
12 REM *               C O P Y P R I N T             *
14 REM * Umwandlung und Ausdruck von Bildschirmcopien *
16 REM *---------------------------------------------*
18 REM *   TI 99/4 A + TI Extended BASIC             *
20 REM * + Externer Speicher (Recorder oder Floppy)  *
22 REM * + Seikosha GP 700 A oder anderen kompartiblen *
24 REM *   Drucker mit 8 DOT Matrix                  *
26 REM *---------------------------------------------*
28 REM * (c) 1984 by Volker Becker                   *
30 REM *     Steinbacher Strasse 10                  *
32 REM *     6370 Oberursel 6                        *
34 REM ***********************************************
36 REM
38 REM
40 REM Drucker Optionen
```

```
42 REM ---------------
44 REM
46 TAB$=CHR$(16)&"025" ! Druckkopf auf Blattmitte positionieren
48 G$=CHR$(27)&"K240" ! Drucker auf 240 Graphiksymbole einstellen
50 GS$=CHR$(27)&"T12" ! Zeilenvorschub auf Graphik einstellen
52 N$=CHR$(27)&"A" ! Zeilenvorschub auf Normalmodus einstellen
54 REM
56 REM ---------------
58 REM
90 DIM GRAPH$(112),BILD$(24),COPY$(24)
100 DISPLAY AT(1,1)ERASE ALL:"   Erstellen der Hardcopy   ----------------------
------"
105 GOTO 5000
110 CALL HCHAR(3,1,32,704)
115 DISPLAY AT(3,1):"Name des CODEFELDES          (GERAET.NAME)"
120 ACCEPT AT(5,1)BEEP:DATEI$
130 ON ERROR 120
140 OPEN #1:DATEI$,INPUT ,SEQUENTIAL,INTERNAL,FIXED 80
145 DISPLAY AT(6,1)BEEP:"--> Codefeld wird eingelesen"
150 FOR I=32 TO 143
160 INPUT #1:GRAPH$(I-32),ASCII$
170 NEXT I
180 CLOSE #1
190 DISPLAY AT(8,1):"Name der Bildschirmdatei     (GERAET.NAME)"
200 ACCEPT AT(10,1)BEEP:DATEI$
210 ON ERROR 200
220 OPEN #1:DATEI$,INPUT ,SEQUENTIAL,INTERNAL,FIXED 80
230 DISPLAY AT(11,1)BEEP:"--> Copyfeld wird eingelesen"
240 FOR I=1 TO 24
250 INPUT #1:BILD$(I)
260 NEXT I
265 CLOSE #1
270 DISPLAY AT(13,1)BEEP:"--> Umwandlung laeuft           BITTE WARTEN"
280 ON ERROR STOP
290 FOR I=1 TO 24
300 COPY$(I)=""
310 FOR K=1 TO 30
320 COPY$(I)=COPY$(I)&GRAPH$(ASC(SEG$(BILD$(I),K,1))-32)
330 NEXT K
340 NEXT I
350 DISPLAY AT(16,1):"Hardcopy abspeichern (J/N)"
360 ACCEPT AT(16,28)SIZE(1)VALIDATE("JN")BEEP:SKP$
370 IF SKP$<>"J" THEN 470
380 DISPLAY AT(17,1):"Name der Hardcopydatei       (GERAET.NAME)"
390 ACCEPT AT(19,1)BEEP:DATEI$
400 IF SEG$(DATEI$,1,3)<>"DSK" THEN 1000
410 ON ERROR 390
420 OPEN #1:DATEI$,OUTPUT,SEQUENTIAL,INTERNAL,FIXED 255
425 ON ERROR STOP
430 FOR I=1 TO 24
440 PRINT #1:COPY$(I)
450 NEXT I
460 CLOSE #1
470 GOSUB 500
480 GOTO 580
500 FOR I=17 TO 19 :: DISPLAY AT(I,1):"" :: NEXT I
510 DISPLAY AT(17,1):"Hardcopy wird ausgedruckt"
520 OPEN #1:"PIO",VARIABLE 255
530 PRINT #1:GS$
540 FOR I=1 TO 24 :: PRINT #1:TAB$;G$;COPY$(I)
550 NEXT I
560 PRINT #1:N$
570 CLOSE #1
```

```
575 RETURN
580 DISPLAY AT(19,1):"Weitere Umwandlungen (J/N)"
590 ACCEPT AT(19,28)SIZE(1)VALIDATE("JN")BEEP:SKP$
600 IF SKP$="J" THEN CALL HCHAR(8,1,32,544):: GOTO 190
610 CALL CLEAR
620 STOP
1000 DISPLAY AT(17,1):"Eine Abspeicherung der       Hardcopy ist nur mit einem D
iskettenlaufwerk moeglich !"
1010 CALL SOUND(250,110,0):: GOTO 350
2000 CALL HCHAR(3,1,32,704)
2005 DISPLAY AT(4,1):"Name der gespeicherten       Hardcopy (GERAET,NAME)"
2010 ACCEPT AT(6,1)BEEP:DATEI$
2020 IF SEG$(DATEI$,1,3)<>"DSK" THEN 3000
2025 ON ERROR 2010
2030 OPEN #1:DATEI$,INPUT ,SEQUENTIAL,INTERNAL,FIXED 255
2035 ON ERROR STOP
2040 FOR I=1 TO 24
2050 INPUT #1:COPY$(I)
2060 NEXT I
2065 CLOSE #1
2070 GOSUB 500
2080 DISPLAY AT(8,1):"Weitere Ausdrucke (J/N)"
2090 ACCEPT AT(8,25)VALIDATE("JN")SIZE(1)BEEP:SKP$
2100 IF SKP$="J" THEN 2000
2110 CALL CLEAR
2120 STOP
3000 DISPLAY AT(8,1):"Nur mit Diskette moeglich !"
3010 CALL SOUND(250,110,0)
3020 GOTO 2010
5000 DISPLAY AT(4,1):"1 - Umwandeln einer Hardcopy2 - Ausdruck  einer Hardcopy"
5010 ACCEPT AT(6,1)SIZE(1)VALIDATE("12")BEEP:M
5020 ON M GOTO 110,2000
5030 END
```



LASER 110 – 4 KByte RAM. Dieser „persönliche Computer" eignet sich als Helfer beim Studium, am Arbeitsplatz oder bei Analysen und Statistiken. Mit seinen Peripheriegeräten erfüllt er alle Ansprüche an einen Home-Computer.

Monster Attack

Commodore 64

Fremde Wesen greifen die Erde an. Ihre Aufgabe besteht darin, die angreifenden Monster vor Erreichen der Erde zu zerstören. Um diese schwierige Aufgabe zu bewältigen, besitzen Sie 3 Abwehrraketen mit unbegrenzter Munition. Das Spiel unterteilt sich in 6 Level, wobei bei jedem Level die Monstergeschwindigkeit sowie die erreichbare Punktzahl ansteigt. Da ein Großteil des Programmes in Maschinensprache geschrieben ist, sollte bei der Eingabe der DATA-Zeilen sehr gewissenhaft vorgegangen werden. Ein Fehler kann zum System-Absturz führen. Das Spiel wird mit Joystick an Port 2 gespielt.



```
1 REM ****************
2 REM  MONSTER ATTACK
3 REM ****************
4 :
10 POKE 53281,0
20 PRINT"{…}"TAB(13)"M O N S T E R"
30 PRINT"{…}"TAB(18)"A T T A C K"
40 PRINT"{…}▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼"
50 PRINT"{…}              (C) BEI F.BRALL   (84)"
100 PRINT"{…}"TAB(14)"BITTE WARTEN"
790 :
800 REM *** MUSIK-NOTEN ***
810 :
815 GOSUB 11000
820 DATA 17,103,250,21,237,250,26,20,400,21,237,100,26,20,25
0,29,69,250
830 DATA 26,20,250,0,0,250,21,237,250,26,20,250,29,69,1000,2
6,20,250,0,0,250
840 DATA -1,-1,0
996 :
997 REM *** MASCHINENPROGRAMM **
998 :
999 FORI= 49152 TO 49787 :READ DA:POKEI,DA:NEXTI
1000 DATA 120,32,26,194,169,18,141,20,3,169,192,141,21,3,88,
76,4
1001 DATA 193,32,186,193,169,224,141,2,220,173,0,220,133,247
,169,255,141
1002 DATA 2,220,165,247,41,16,240,110,165,247,41,4,240,9,165
,247,41
1003 DATA 8,240,33,76,49,234,173,62,3,208,5,173,63,3,240,243
,56
1004 DATA 173,62,3,233,2,141,62,3,173,63,3,233,0,141,63,3,76

1005 DATA 119,192,173,63,3,240,7,173,62,3,201,80,16,211,24,1
73,62
1006 DATA 3,105,2,141,62,3,173,63,3,105,0,141,63,3,76,119,19
2
1007 DATA 173,62,3,141,0,208,173,63,3,240,11,173,16,208,9,1,
141
1008 DATA 16,208,76,54,192,173,16,208,41,254,141,16,208,76,5
4,192,165
1009 DATA 2,240,3,76,54,192,173,62,3,141,6,208,173,63,3,240,
11
1010 DATA 173,16,208,9,8,141,16,208,76,189,192,173,16,208,41
,247,141
1011 DATA 16,208,169,2,141,1,212,169,0,141,4,212,169,15,141,
24,212
```




```
1012 DATA 169,9,141,5,212,169,131,141,4,212,32,84,194,162,22
0,142,7
1013 DATA 208,32,84,194,169,1,133,2,76,54,192,120,165,2,208,
2,88
1014 DATA 96,206,7,208,32,218,193,32,84,194,173,7,208,240,2,
88,96
1015 DATA 169,0,133,2,96,173,21,3,201,234,208,1,96,169,0,141
,30
1016 DATA 208,141,30,208,32,98,194,165,162,41,1,240,43,162,4
0,173,61
1017 DATA 3,240,7,173,60,3,201,80,16,29,24,173,60,3,105,1,14
1
1018 DATA 60,3,173,61,3,105,0,141,61,3,32,113,193,32,154,193
,202
1019 DATA 208,218,76,4,193,162,40,173,60,3,208,5,173,61,3,24
0,201
1020 DATA 56,173,60,3,233,1,141,60,3,173,61,3,233,0,141,61,3
1021 DATA 32,113,193,32,154,193,202,208,220,76,69,193,120,17
3,60,3,141
1022 DATA 2,208,173,61,3,240,14,173,16,208,9,2,141,16,208,16
9,0
1023 DATA 141,30,208,96,173,16,208,41,253,141,16,208,169,0,1
41,30,208
1024 DATA 88,96,72,138,72,152,72,166,250,160,150,165,162,153
,160,216,153
1025 DATA 148,217,136,208,245,32,232,192,202,208,237,104,168
,104,170,104,96
1026 DATA 198,248,240,1,96,238,3,208,173,3,208,201,220,240,5
,165,249
1027 DATA 133,248,96,120,169,49,141,20,3,169,234,141,21,3,96
,120,173
1028 DATA 30,208,240,6,41,10,201,10,240,4,32,84,194,96,169,0
,141
1029 DATA 4,212,169,6,141,1,212,169,15,141,24,212,169,10,141
,5,212
1030 DATA 169,0,141,6,212,169,131,141,4,212,169,0,141,30,208
,169,3
1031 DATA 141,249,7,169,2,141,33,208,76,206,193,169,0,141,16
,208,141
1032 DATA 63,3,133,2,169,60,141,3,208,165,162,141,60,3,169,0
,141
1033 DATA 61,3,141,26,208,32,113,193,169,90,141,62,3,141,0,2
08,165
1034 DATA 249,133,248,169,14,141,249,7,169,255,141,7,208,169
,6,141,32
1035 DATA 208,169,0,141,26,208,169,0,141,25,208,173,30,208,9
6,169,250
1036 DATA 141,8,212,169,0,141,11,212,169,8,141,12,212,169,4,
141,13
1037 DATA 212,169,19,141,11,212,96
2000 :
2010 REM SPRITE 0 EINLESEN
2020 :
2030 FORI=832 TO 894:READ DA:POKE I,DA:NEXTI
2040 DATA 0,20,0,0,20,0,0,20,0
2044 DATA 32,20,8,8,20,32,8,20,32
2046 DATA 2,105,128,0,105,0,0,105,0
2048 DATA 0,105,0,0,105,0,0,65,0
2050 DATA 0,65,0,1,170,64,1,170,64
2052 DATA 6,170,144,6,170,144,21,85,84
2054 DATA 0,85,0,1,85,64,1,85,64
2060 :
```

```
2070 REM SPRITE 1 EINLESEN
2080 :
2090 FORI=896 TO 958:READ DA:POKE I,DA:NEXTI
2100 DATA 63,0,252,0,195,0,0,60,0
2110 DATA 2,170,128,10,170,160,10,170,160
2120 DATA 42,170,168,41,105,104,42,170,168
2130 DATA 42,150,168,42,170,168,42,170,168
2140 DATA 41,170,104,42,85,168,10,170,160
2150 DATA 10,170,160,2,170,128,2,0,128
2160 DATA 2,0,128,8,0,32,32,0,8
2200 :
2290 REM SPRITE 3 EINLESEN
2300 :
2305 FORI=704 TO 766:READ DA:POKE I,DA:NEXT
2310 DATA 0,0,0,0,0,0,0,0,0
2315 DATA 0,0,0,0,0,0,0,0,0
2320 DATA 0,80,0,1,84,0,1,84,0
2325 DATA 1,84,0,1,84,0,0,80,0
2330 DATA 0,0,0,0,0,0,0,0,0
2340 DATA 0,0,0,0,0,0,0,0,0
2345 DATA 0,0,0,0,0,0,0,0,0
2400 :
2410 REM *** SPRITES VORBEREITEN
2420 :
2425 POKE 2040,13:POKE 2041,14:POKE 2042,15:POKE 2043,11
2430 POKE 53269,15:REM SPRITES ON
2435 POKE 53276,15:POKE 53285,3:POKE 53286,6:POKE 53287,2
2440 POKE 53249,227
3000 :
3001 REM *** STERNE ZEICHNEN ***
3002 :
3010 PRINT "[illegible]RAKETEN:3  PUNKTE:0      LEVEL:1"
3012 PRINT"[illegible]";
3015 FORI=1 TO 100
3020 W=INT(RND(1)*800)+1104
3025 POKE W,46:NEXT
3030 FORI=55376 TO 56200:POKE I,PEEK(162):NEXT
4000 :
4001 REM *** HAUPTPROGRAMM ***
4002 :
4005 GOSUB 11000
4010 AR=3:TA=5:TR=5:R=3:L=1:ZA=0
4012 POKE 53281,0
4020 POKE 250,TA:POKE 249,TR
4025 POKE 2040,13:POKE 53251,0:POKE 53277,0
4030 POKE 53286,L:SYS49152
4040 IF PEEK (2041)<>14 THEN 10000
4050 :
4055 REM *** MONSTER GELANDET ***
4060 :
4065 POKE 53277,2
4070 MP=PEEK(828)+(PEEK(829)*256)
4080 RP=PEEK(830)+(PEEK(831)*256)
4090 S=1:IF MP>RP THEN S=-1
4100 FORI=MP TO RP STEP S
4110 HB=INT(I/256):LB=I-(HB*256)
4115 POKE 53250,LB
4120 IF HB=0 THEN POKE 53264,(PEEK(53264)AND 1)
4130 IF HB=1 THEN POKE 53264,(PEEK(53264)OR  2)
4140 NEXT:POKE 2040,3
```

```
4150 POKE 54273,10:POKE 54276,0:POKE 54296,15:POKE 54277,12:
POKE 54278,0
4160 POKE 54276,131
4170 FORI=1 TO 10:POKE 53281,I:FORQ=1 TO 30:NEXTQ:NEXTI
4180 POKE 53281,0
4190 R=R-1
5010 IF R<0 THEN 12000
5020 GOTO 10090
10000 :
10001 REM *** MONSTER GETROFFEN ***
10002 :
10010 FORI=1 TO 50
10020 POKE 53281,I
10030 NEXTI:POKE 53281,0
10040 SC=SC+(L*100):ZA=ZA+1
10050 IF L=6 THEN 10090
10060 IF ZA=5 THEN ZA=0:TA=TA-1:L=L+1:PRINT"███ NEXT LEVEL ▬
     ":GOSUB 11000
10070 IF TA<1 THEN TA=1:TR=TR-1
10080 IF TA<1 THEN TA=1
10090 PRINT"█                                          "
10100 PRINT"██ RAKETEN:"R"  PUNKTE:"SC"  LEVEL:"L
10110 PRINT"████████████████████████ ___ ____ _____
_______ _ _";
10120 GOTO4020
```

Musik-Untermalung

```
11000 :
11010 REM *** MUSIK SPIELEN ***
11020 :
11030 RESTORE
11040 SI=54272:FL=SI:FH=SI+1:TL=SI+2:TH=SI+3:W=SI+4:A=SI+5:H
=SI+6:L4=SI+24
11050 POKE L4,15:POKE TH,13:POKE TL,15:POKE A,3*16+15:POKE H
,9
11060 READ X:READ Y:READ D
11070 IF X=-1 THEN RETURN
11080 POKE FH,X:POKE FL,Y
11090 POKE W,65
11100 FOR T=1 TO D:NEXT
11110 POKE W,0:GOTO 11060
12000 :
12010 REM *** GAME OVER ***
12020 :
12030 PRINT"███████"TAB(08)"█ *** GAME OVER *** "
12040 IF SC<1000 THENPRINT"█"TAB(5)"VIEL SCHLECHTER GING ES
NICHT !":GOTO 12100
12050 IF SC<5000 THENPRINT"█"TAB(2)"FUER DAS ERSTE SPIEL NIC
HT SCHLECHT !":GOTO12100
12060 IF SC<10000 THENPRINT"█"TAB(5)"AUS DIR KANN NOCH WAS W
ERDEN !":GOTO 12100
12070 PRINT"█"TAB(3)"GRATULIERE ! DAS WAR WIRKLICH GUT !"
12100 PRINT"██  NOCH EIN SPIEL (J/N) "
12110 GET E$:IF E$=""THEN 12110
12120 IF E$="J"THEN RUN
12130 PRINT"█":END
NREADY.
```

Block Painter

Eine der Stärken des Commodore 64 besteht in der Möglichkeit einen eigenen Zeichensatz zu entwerfen. Dies ist für das Programmieren von Spielen sehr günstig und hat gegenüber der hochauflösenden Grafik den Vorteil, das wesentlich weniger Speicherplatz benötigt wird. Dies klingt alles sehr schön, ist jedoch in der Praxis nur den Profi-Programmierern möglich, da auch Maschinenroutinen erforderlich sind, um den Zeichensatz zu ändern.
Eine weitere Schwierigkeit besteht im Aufbau der Bildschirm-Grafik. Will man beispielsweise in einer „Print-Zeile" oft die Farbe wechseln oder einige Reverse-Zeichen benutzen, so wird die Zeile häufig durch Kontrollzeichen unübersichtlich oder völlig unmöglich. Ganz zu schweigen davon, daß man wissen muß, welches Zeichen auf welcher Taste liegt.
Diesen Nachteilen begegnet der hier vorgestellte Block-Painter. Die Bedienung des Programmes ist vorbildlich gelöst worden. Während der Erstellung wird kein einziges Mal die Tastatur benötigt, da alle Funktionen mit Joystick 2 gesteuert werden.

Anwendung:
Nach dem Starten des Programmes erscheint etwa in der Mitte des Bildschirmes ein blinkender Pfeil, umgeben von zahlreichen Befehlsworten. Möchten Sie beispielsweise das Zeichen „E" verändern, so steuern Sie Ihren Pfeil auf dieses Zeichen und drücken die Taste. Wie Sie erkennen werden, hängt nun das „E" an Ihrem Pfeil. Bringen Sie dieses zum Befehl „EDIT" und betätigen nochmals die Taste. Nun wird das Zeichen Ihrem Pfeil entnommen und in einem großen Fenster angezeigt. Sie können nun mit Hilfe Ihres Pfeiles (Joystick) beliebige Punkte setzen. Möchten Sie Punkte löschen, so fahren Sie mit Ihrem Pfeil auf das Befehlswort „CLEAR". Nun sind Sie im Clear-Modus und können beliebige Punkte löschen. Durch „SET" kommen Sie wieder in den ursprünglichen Modus zurück. Möchten Sie Ihr Zeichen invertieren, so benutzen Sie den „RVS"-Befehl. Ist Ihr Zeichen erstellt, gehen Sie auf den „COPY"-Befehl und anschließend auf das Zeichen, welches Sie ersetzen möchten. Sie haben nun Ihr eigenes Zeichen entworfen. Mit dem Befehl „TABLE" können Sie zwischen GRAFIK und KLEINSCHRIFT umschalten. Der Befehl „ORIGINAL" sollte mit Vorsicht behandelt werden, da er den ursprünglichen Zeichensatz wieder aktiviert.
Um nun ein Bild zu entwerfen, gehen Sie zuvor auf den Befehl „ERASE-PICTURE", um den Grafik-Speicher zu löschen. Holen Sie mit dem Pfeil ein beliebiges Zeichen aus dem Zeichensatz und fahren Sie auf eine der Farben des Befehles „COLOR". Dieser wählt die Zeichenfarbe; der Befehl „GROUND" die Hintergrundfarbe. Wenn sie nun Ihren Pfeil über die untere oder obere Bildschirmgrenze bewegen, so schaltet der Computer auf die Grafikseite um. Sie können Ihr Zeichen nun beliebig oft an jene beliebige Stelle zeichnen. Möchten Sie eine andere Farbe oder ein anderes Zeichen zeichnen, so überschreiten Sie wieder die Bildschirmgrenze. Beachten Sie, daß im Clear-Modus einzelne Zeichen gelöscht werden können. Durch den Befehl „COLOR-EDIT" können einzelne Zeichen in Ihrer Farbe geändert werden.

Mit dem Befehl „PICTURE-SAVE" wird das Bild auf Disk abgespeichert. Wenn Sie die letzten 4 Zeichen („,8") in Zeile 1860 weglassen, so wird das Bild auf Cassette gespeichert und das Bild in beiden Fällen mit neuem Zeichensatz und kleinem Maschinenprogramm, insgesamt ca. 6k, abgespeichert. Geladen wird das Bild einfach mit LOAD „name",8,1. Durch den Befehl „SYS 34704" wird das Bild gezeichnet. Das „Bild" belegt die Speicherstellen 34707–40960. Nach dem „SYS-Aufruf" kann dieser Bereich wieder gelöscht werden.

Hat man sich erst einmal an die etwas ungewohnte Steuerung gewöhnt, dann werden Sie bald die Möglichkeiten des Programmes zu schätzen wissen.

```
10 REM ####################
20 REM # BLOCK - PAINTER  #
30 REM ####################
40 REM
50 REM
60 POKE 55,3:POKE56,48:REM ** HIMEM **
70 POKE 53281,1:POKE 53280,2:REM FARBE
80 POKE 53280,6
90 WE=6:MI=255:MA=0
100 REM ** ZEICHENSATZ VERSCHIEBEN **
110 REM ** NACH $3000 = 12288 **
120 FORI=51200 TO 51252
130 READ DA:POKE I,DA:NEXTI:SYS 51200
140 DATA 120,165,1,72,41,251,133,1,169,208,133,3
150 DATA 169,48,133,5,160,0,132,2,132,4,162,32
160 DATA 177,2,145,4,200,208,249,230,3,230,5,202
170 DATA 208,242,104,133,1,173,24,208,41,241,9,12
180 DATA 141,24,208,88,96
190 REM ** SCREEN-SWITCH **
200 FORI =51253 TO 51409:READ DA:POKEI,DA:NEXTI
210 DATA 169,216,162,219,160,75,32,203,200,169,0,162,232,160,232,32,122
220 DATA 200,169,4,162,7,160,71,32,203,200,169,0,162,232,160,232,32
230 DATA 122,200,169,88,162,91,160,219,32,203,200,169,0,162,232,160,232
240 DATA 32,122,200,169,84,162,87,160,7,32,203,200,169,0,162,232,160
250 DATA 232,133,95,134,90,132,88,76,191,163,169,216,162,219,160,91,32
260 DATA 203,200,169,0,162,232,160,232,32,122,200,169,4,162,7,160,87
270 DATA 32,203,200,169,0,162,232,160,232,32,122,200,169,72,162,75,160
280 DATA 219,32,203,200,169,0,162,232,160,232,32,122,200,169,68,162,71
```

```
290 DATA 160,7,32,203,200,169,0,162,232,160,232,76,122,200,133,96,134
300 DATA 91,132,89,96
310 REM ** SPRITE - BEFEHLSZEIGER **
320 FORI=704 TO767
330 READ DA:POKE I,DA:NEXTI
340 POKE 2040,11:POKE 53287,0:POKE 53269,1:X%=180:Y%=100
350 DATA 7,192,0,7,128,0,7,128,0
360 DATA 7,128,0,4,192,0,0,96,0
370 DATA 0,48,0,0,24,0,0,12,0
380 DATA 0,0,0,0,0,0,0,0,0
390 DATA 0,0,0,0,0,0,0,0,0
400 DATA 0,0,0,0,0,0,0,0,0
410 DATA 0,0,0,0,0,0,0,0,0
420 GOSUB 770:REM ** MASKE **
430 GOSUB 960:REM ** ZEICHENSATZ **
440 REM ** JOYSTICK EINGABE **
450 POKE56322,224:J=PEEK(56320)
460 IF(JAND1)=0THENY%=Y%-3
470 IF(JAND2)=0THENY%=Y%+3
480 IF(JAND4)=0THENX%=X%-3
490 IF(JAND8)=0THENX%=X%+3
500 IF X%>335 THEN X%=20
510 IF X%<20 THEN X%=335
520 IF Y%>245 THEN 1480
530 IF Y%<30 THEN 1480
540 HB=INT(X%/256):LB=(X%AND255)
550 POKE 53269,0:POKE 53248,LB:POKE 53249,Y%:POKE 53264,HB:POKE 53269,1
560 IF(JAND16)=0 THEN 580
570 GOTO 450
580 REM ** BEFEHL FESTSTELLEN **
590 POKE56322,255
600 AD=INT((Y%-50)/8)*40 +INT((X%-20)/8)+1024
610 IF X%<100 AND X%>36 AND Y%>74 AND Y%<138 THEN 1210
620 IF X%>208 AND Y%<200 THEN 1000:REM UEBERNAHME
630 IF AD>1235 AND AD<1243 THEN POKE 53280,6:WE=6:GOTO450
640 IF AD>1155 AND AD<1163 THEN POKE 53280,8:WE=1:GOTO450
650 IF AD>1315 AND AD<1323 THEN 1140
660 IF AD>1395 AND AD<1403 THEN 1230
670 IF AD>1475 AND AD<1483 THEN 1310
680 IF AD>1555 AND AD<1563 THEN 1390
690 IF AD>1544 AND AD<1555 THEN SYS51200:TA=0:GOTO 450
700 IF AD>1871 AND AD<1904 THEN 1420
710 IF AD>1951 AND AD<1984 THEN 1450
720 IF AD>1624 AND AD<1640 THEN FORI=21504 TO 22505:POKEI,32:NEXTI:GOTO450
730 IF AD>1704 AND AD<1720 THEN 1770
740 IF AD>1784 AND AD<1799 THEN 1790
750 GOTO450 :REM ** ZUR EINGABE **
760 REM ** BILDSCHIRMMASKE **
770 PRINT"    >>> BLOCK - PAINTER <<<               "
780 PRINT"                          "
790 PRINT "                          CLEAR  "
800 PRINT "                          "
810 PRINT "                          SET    "
820 PRINT "                          "
830 PRINT "                          EDIT   "
840 PRINT "                          "
850 PRINT "                          RVS    "
860 PRINT "                          "
```

```
870 PRINT "          COPY  "
880 PRINT:PRINT"  ORIGINAL   TABLE "
890 PRINT"  ERASE PICTURE "
900 PRINT"  COLOR EDIT     "
910 PRINT"  PICTURE-SAVE  "
920 PRINT"  GROUND                                "
930 PRINT"  COLOR                                ";
940 RETURN
950 REM ** ZEICHENSATZ ANZEIGEN **
960 Z=0
970 FORU=0 TO16:P=U*40
980 FORI=1128+P TO 1142+P:POKE I,Z:Z=Z+1:POKE I+54272,14:NEXT I,U
990 RETURN
1000 REM ** ZEICHEN UEBERNEHMEN **
1010 IF PEEK(1483)=31 THEN 1060
1020 PO=(8*PEEK(AD))+12288+TA:ZE=PEEK(AD)
1030 POKE 1719,160
1040 FORI=0 TO 7:POKE 741+I*3,PEEK(PO+I)
1050 NEXTI:GOTO450
1060 REM ** ZEICHEN UEBERGEBEN **
1070 POKE 1483,32
1080 PO=(8*PEEK(AD))+12288+TA
1090 I=PEEK(AD):IF TA<>0THEN I=I+256
1100 IF I<MI THEN MI=I
1110 IF I>MA THEN MA=I
1120 FORI=0 TO 7:POKE 741+I*3,0:POKEPO+I,CD(I):NEXTI
1130 GOTO 450
1140 REM ** ZEICHEN VERGROESSERN **
1150 POKE1483,32:PRINT"      ";
1160 FORI=0 TO 7:Z=PEEK(741+(I*3)):POKE 741+(I*3),0
1170 FORJ=7 TO 0 STEP-1
1180 A=Z AND 2↑J:IF A THEN PRINT"  ";:GOTO 1200
1190 PRINT"  ";
1200 NEXT:PRINT:PRINT"   ";:NEXT:GOTO 450
1210 REM ** PUNKT SETZEN/LOESCHEN **
1220 POKE AD,160:POKE AD+54272,WE:GOTO450
1230 REM ** ZEICHEN IN RVS WANDELN **
1240 FORI=0 TO 7
1250 FORJ=0 TO 7:C=PEEK(55418+J+(I*40))AND 15
1260 IF C=6 THEN C=1:GOTO1280
1270 C=6
1280 POKE 55418+J+(I*40),C
1290 POKE 1146+J+(I*40),160
1300 NEXT J,I:GOTO 450
1310 REM ** COPIERE IN ZEICHENSATZ **
1320 FORI=0 TO 7
1330 CD(I)=0:U=8
1340 FORJ=0 TO 7:U=U-1
1350 C=PEEK(55418+J+(I*40))AND15
1360 IF C=6 THEN CD(I)=CD(I)+(2↑U)
1370 NEXT J:POKE 741+(I*3),CD(I):NEXTI
1380 POKE 1483,31:POKE1483+54272,0:GOTO450
1390 REM ** ZEICHENSATZ WECHSELN **
1400 IF TA=0 THEN TA=2048:PRINTCHR$(14);:GOTO 450
1410 IF TA=2048 THEN TA=0:PRINTCHR$(142);:GOTO450
1420 REM ** HINTERGRUNDFARBE **
1430 BG=PEEK(AD+54272):POKE 1864+54272,BG
1440 POKE 1865+54272,BG:GOTO450
```

```
1450 REM ** ZEICHENFARBE **
1460 CO=PEEK(AD+54272):POKE 1944+54272,CO
1470 POKE 1945+54272,CO:GOTO450
1480 REM ** BILD-EDITOR **
1490 X%=190:Y%=60:BG=BGAND15:MS=PEEK(53287):POKE 53287,0
1500 IF BG=0 OR BG=6 OR BG=11 OR BG=9 OR BG=2 THEN POKE 53287,1
1510 POKE53281,BG:SYS 51253
1520 REM ** JOYSTICK EINGABE **
1530 POKE56322,224:J=PEEK(56320)
1540 IF(JAND1)=0THENY%=Y%-3
1550 IF(JAND2)=0THENY%=Y%+3
1560 IF(JAND4)=0THENX%=X%-3
1570 IF(JAND8)=0THENX%=X%+3
1580 IF X%>335 THEN X%=20
1590 IF X%<20 THEN X%=335
1600 IF Y%>250 THEN 1660
1610 IF Y%<30 THEN 1660
1620 HB=INT(X%/256):LB=(X%AND255)
1630 POKE 53269,0:POKE 53248,LB:POKE 53249,Y%:POKE 53264,HB:POKE 53269,1
1640 IF(JAND16)=0 THEN 1690
1650 GOTO 1530
1660 REM ** ZURUECK INS MENUE **
1670 X%=190:Y%=210:POKE53281,1:SYS 51331:POKE53287,MS
1680 GOTO450
1690 REM ** ZEICHEN SETZEN **
1700 POKE56322,255
1710 AD=INT((Y%-50)/8)*40 +INT((X%-20)/8)+1024
1720 IF(PEEK(53280)AND15) =8 THEN ZD=32:GOTO1740
1730 ZD=ZE:IF ZD=256 THEN 1750
1740 POKE AD,ZD
1750 POKE AD+54272,CO
1760 GOTO 1530
1770 REM ** COLOR EDIT **
1780 ZE=256:FORI=0 TO 7:POKE 741+(I*3),0:NEXTI:POKE 1719,31:GOTO450
1790 REM ** DATA-GENERATOR **
1800 PRINT" AUGENBLICK BITTE ! "
1810 FORI= 34704 TO 34803 :READ DA:POKEI,DA:NEXTI
1820 FORI=21504 TO23552:POKEI+13312,PEEK(I):NEXTI
1830 FORI=12288 TO16383:POKEI+24576,PEEK(I):NEXTI
1840 PRINT"FILENAME  ";:INPUT F$
1850 PRINT"POKE 43,144:POKE44,135 :POKE45,0:POKE46,160"
1860 PRINT"SAVE"CHR$(34)"@:"F$CHR$(34)",8"
1870 POKE631,19:FORI=632TO640:POKEI,13:NEXT:POKE198,9
1880 DATA 169,0,133,95,169,136,133,96,169,232,133,90,169,139,133,91,169
1890 DATA 232,133,88,169,195,133,89,32,191,163,169,0,133,95,169,140,133
1900 DATA 96,169,232,133,90,169,143,133,91,169,232,133,88,169,219,133,89
1910 DATA 32,191,163,169,0,133,95,169,144,133,96,169,0,133,90,169,160
1920 DATA 133,91,169,0,133,88,169,240,133,89,32,191,163,169,192,141,136
1930 DATA 2,169,8,141,24,208,173,0,221,41,252,141,0,221,96
```

Das hier vorgestellte Maschinenprogramm erlaubt den Anschluß eines MX80 oder MX80 Typ 3 an den User-Port des Commodore 64. Obwohl kein Interface benötigt wird, gestattet das Programm den Ausdruck aller Commodore Zeichen inclusive Grafik. Sicher wird sich so mancher überlegen wie es möglich ist, Commodore Zeichen auf einem Epson-Drucker auszugeben, da diese bekanntlich keine Grafik-Zeichen erlauben. Die Erklärung ist sehr einfach: Die Epson Drucker MX80 Typ 3 und MX82 besitzen die Möglichkeit, 8 vertikal angeordnete Nadeln einzeln anzusteuern. Das hier vorgestellte Programm nutzt diese Möglichkeit aus, indem es mit Hilfe eines Zeichensatzes das auszugebende Zeichen in 8 mal 8 Nadel-Stellungen an den Drucker sendet. Um das Eingeben eines 4 kByte Zeichensatzes zu ersparen, benutzt das Programm den Zeichensatz des Bildschirmspeichers. Da mit einer 8×8 Matrix gearbeitet wird, beträgt die normale Zeichenzahl 60 Zeichen pro Zeile. Dies reicht für die meisten Anwendungen völlig aus. Werden jedoch einmal mehr Zeichen benötigt, so kann mit der Sekundäradresse 1 auf 120 Zeichen pro Zeile umgeschaltet werden. Das Programm belegt nur 604 Bytes und liegt im Bereich $C000 - $C25B und ist somit nicht durch Basic zu überschreiben. Nach dem Laden wird es mit SYS49152 initialisiert und kann danach wie gewohnt mit OPEN angesteuert werden.

Beispiele:
OPEN 1,4,0 : CMD1 : LIST : REM Basic-Listing in Normalschrift ausgeben
OPEN 1,4,1 : CMD1 : LIST : REM Basic-Listing in Schmalschrift ausgeben
OPEN 1,4,2 : CMD1 : LIST : REM Basic-Listing in Fettschrift (60Z) ausgeben
Indem man 3 Files mit verschiedenen Sekundäradressen öffnet, können auch mehrere Schriftarten gleichzeitig benutzt werden. In einer Zeile ist jedoch immer nur 1 Schriftart möglich. Zum Anschluß des Druckers an den User-Port benötigt man ein 11- oder mehrpoliges Kabel sowie 2 geeignete Stecker. Die Kontaktbelegung lautet wie folgt:

USER PORT	EPSON DRUCKER	
A	GND	16
B	Flag-Busy	11
C	Data 1	2
D	Data 2	3
E	Data 3	4
F	Data 4	5
H	Data 5	6
J	Data 6	7
K	Data 7	8
L	Data 8	9
M	PA2-Strobe	1

Hinweis:
Für den fortgeschrittenen C-64-Fan ist das Maschinenprogramm als Assembler-Listing abgedruckt!

```
10 PRINT"DIESES LISTING WURDE AUF"
20 PRINT"EINEM MX 82 AUSGEDRUCKT !"
30 PRINT"EIN INTERFACE WIRD NICHT BENOETIGT"
40 PRINT"GRAFIK: U.S.W."
50 PRINT" "
60 PRINT" "
```


```
10 PRINT"DIESES LISTING WURDE AUF"
20 PRINT"EINEM MX 82 AUSGEDRUCKT !"
30 PRINT"EIN INTERFACE WIRD NICHT BENOETIGT"
40 PRINT"GRAFIK: U.S.W."
50 PRINT" "
60 PRINT" "
```

```
10 PRINT"DIESES LISTING WURDE AUF"
20 PRINT"EINEM MX 82 AUSGEDRUCKT !"
30 PRINT"EIN INTERFACE WIRD NICHT BENOETIGT"
40 PRINT"GRAFIK: U.S.W."
50 PRINT" "
60 PRINT" "
```


Basic-Loader

```
10 FORI= 49152 TO 49757 :READ DA:POKEI,DA:NEXTI
60000 DATA 169,88,160,192,141,26,3,140,27,3,169,139,160,192,141,28,3
60001 DATA 140,29,3,169,163,160,192,141,30,3,140,31,3,169,186,160,192
60002 DATA 141,32,3,140,33,3,169,209,160,192,141,38,3,140,39,3,169
60003 DATA 255,141,3,221,173,2,221,9,4,141,2,221,96,141,1,221,169
60004 DATA 16,44,13,221,240,251,173,0,221,9,4,141,0,221,41,251,141
60005 DATA 0,221,96,166,184,240,5,32,15,243,208,3,76,254,246,166,152
60006 DATA 224,10,144,3,76,251,246,230,152,165,184,157,89,2,165,185,9
60007 DATA 96,157,109,2,165,186,157,99,2,201,4,208,2,24,96,201,0
60008 DATA 76,119,243,32,20,243,240,2,24,96,32,31,243,138,72,165,186
60009 DATA 201,4,240,3,76,157,242,76,241,242,32,15,243,240,3,76,1
60010 DATA 247,32,31,243,165,186,201,4,208,3,76,10,247,76,25,242,32
60011 DATA 15,243,240,3,76,1,247,32,31,243,165,186,201,4,208,3,76
60012 DATA 117,242,76,91,242,72,165,154,201,4,240,3,76,205,241,165,185
60013 DATA 41,15,201,0,208,23,169,60,141,67,193,169,75,141,68,193,169
```

```
60014 DATA 0,141,69,193,169,8,141,70,193,76,182,193,201,1,208,23,169
60015 DATA 120,141,67,193,169,76,141,68,193,169,0,141,69,193,169,8,141
60016 DATA 70,193,76,182,193,201,2,208,23,169,60,141,67,193,169,76,141
60017 DATA 68,193,169,1,141,69,193,169,16,141,70,193,76,182,193,76,227
60018 DATA 192,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
60019 DATA 0,0,0,0,141,65,193,72,152,72,138,72,174,65,193,169,0
60020 DATA 133,247,173,24,208,41,2,24,10,10,9,208,133,248,224,0,240
60021 DATA 16,24,165,247,105,8,133,247,165,248,105,0,133,248,202,208,240
60022 DATA 165,1,141,59,193,169,49,120,133,1,160,0,162,7,177,247,106
60023 DATA 62,51,193,202,16,249,200,192,8,208,240,173,59,193,133,1,88
60024 DATA 162,0,189,51,193,72,32,64,192,104,172,69,193,240,3,32,64
60025 DATA 192,232,224,8,208,235,104,170,104,168,104,24,96,104,141,65,193
60026 DATA 72,152,72,138,72,173,65,193,201,13,208,84,169,0,141,63,193
60027 DATA 141,64,193,174,62,193,240,58,24,173,63,193,109,70,193,141,63
60028 DATA 193,173,64,193,105,0,141,64,193,202,208,235,169,27,32,64,192
60029 DATA 173,68,193,32,64,192,173,63,193,32,64,192,173,64,193,32,64
60030 DATA 192,162,0,189,60,3,32,71,193,232,236,62,193,208,244,169,13
60031 DATA 32,64,192,169,0,141,62,193,76,175,193,173,0,4,141,66,193
60032 DATA 165,214,141,60,193,165,211,141,61,193,32,102,229,173,65,193,32
60033 DATA 22,231,173,0,4,174,62,193,157,60,3,173,66,193,141,0,4
60034 DATA 174,60,193,172,61,193,24,32,10,229,174,62,193,232,142,62,193
60035 DATA 236,67,193,208,3,76,198,193,76,175,193

READY.
```

```
                              ;
                              ; ***** C 64 EPSON-PRINTER   *****
                              ; C64 USER-PORT TO  PRINTER MX 82
                              ;
                              ; INIT MIT SYS 12 * 4096
                              ; SEKUNDAERADRESSE 0 = 60 ZEICHEN
                              ; SEKUNDAERADRESSE 1 =120 ZEICHEN
                              ; SEKUNDAERADRESSE 2 = 60 ZEICHEN FETTSCHRIFT
120:    C000                         .OPT P1,OO
                              ;
140:    C000                TABLE    =     53248    ;ZEICHENSATZ AB $D000
150:    C000                PUFFER   =     828      ;KASSETTENPUFFER ALS DATENPUFFER
                              ;
170:    C000                OPENV    =     $031A    ;OPEN-VECTOR
180:    C000                CLOSEV   =     $031C    ;CLOSE-VECTOR
190:    C000                CHKINV   =     $031E
200:    C000                CHKOUTV  =     $0320
210:    C000                BSOUTV   =     $0326    ;ZEICHENAUSGABE-VECTOR
220:    C000                XREG     =     $97      ;SPEICHER FUER REGISTER
230:    C000                TABA     =     $F7      ;POINTER
                             ;CIA 2
250:    C000                PORTA    =     56576    ;PORT A
260:    C000                PORTB    =     56577    ;PORT B (USER PORT)
270:    C000                DRRA     =     56578    ;DATENRICHTUNG A
280:    C000                DRRB     =     56579    ; " USER PORT
290:    C000                ICR      =     56589    ;INTERRUPT CONTROL REG.
300:    C000                LF       =     $B8      ;FILENUMMER
310:    C000                SA       =     $B9      ;SEKUNDAERADRESSE
320:    C000                FA       =     $BA      ;GERAETENR.
330:    C000                NMBFILES =     $98      ;ANZAHL DER OFFENEN FILES
340:    C000                LFTAB    =     $259
350:    C000                FATAB    =     $263
360:    C000                SATAB    =     $26D
370:    C000                SRCHFIL  =     $F30F    ;SUCHT FILENR.
```

```
380:   C000
390:   C000             INIT      *=    $C000    ;PROGRAMMANFANG
400:   C000 A9 58                 LDA   #<OPEN   ;OPEN-VECTOR UMSTELLEN
410:   C002 A0 C0                 LDY   #>OPEN
420:   C004 8D 1A 03              STA   OPENV
430:   C007 8C 1B 03              STY   OPENV+1
440:   C00A A9 8B                 LDA   #<CLOSE   ;CLOSE-VECTOR UMSTELLEN
450:   C00C A0 C0                 LDY   #>CLOSE
460:   C00E 8D 1C 03              STA   CLOSEV
470:   C011 8C 1D 03              STY   CLOSEV+1
480:   C014 A9 A3                 LDA   #<CHKIN   ;EINGABEGERAET SETZEN
490:   C016 A0 C0                 LDY   #>CHKIN
500:   C018 8D 1E 03              STA   CHKINV
510:   C01B 8C 1F 03              STY   CHKINV+1
520:   C01E A9 BA                 LDA   #<CHKOUT   ;AUSGABEGERAET SETZEN
530:   C020 A0 C0                 LDY   #>CHKOUT
540:   C022 8D 20 03              STA   CHKOUTV
550:   C025 8C 21 03              STY   CHKOUTV+1
560:   C028 A9 D1                 LDA   #<BSOUT   ;ZEICHEN-AUSGABE UMSTELLEN
570:   C02A A0 C0                 LDY   #>BSOUT
580:   C02C 8D 26 03              STA   BSOUTV
590:   C02F 8C 27 03              STY   BSOUTV+1
600:   C032 A9 FF                 LDA   #$FF
610:   C034 8D 03 DD              STA   DRRB     ;PORT B AUF AUSGABE
620:   C037 AD 02 DD              LDA   DRRA
630:   C03A 09 04                 ORA   #%100
640:   C03C 8D 02 DD              STA   DRRA     ;PA2 AUF AUSGABE
650:   C03F 60                    RTS
                          ;
                          ; AUSGABE MIT HANDSHAKE
                          ; DATEN AUF PORT B
                          ; STROBE AUF PA2
                          ; BUSY UEBER FLAG NACH ICR
                          ;
720:   C040 8D 01 DD AUSGABE      STA   PORTB    ;DATEN AUSGEBEN
730:   C043 A9 10                 LDA   #%10000   ;MASKE FUER FLAG BIT
740:   C045 2C 0D DD TESTBUSY     BIT   ICR
750:   C048 F0 FB                 BEQ   TESTBUSY
760:   C04A AD 00 DD              LDA   PORTA
770:   C04D 09 04                 ORA   #%100    ;STROBE SETZEN
780:   C04F 8D 00 DD              STA   PORTA
790:   C052 29 FB                 AND   #%11111011   ;STROBE LOESCHEN
800:   C054 8D 00 DD              STA   PORTA
810:   C057 60                    RTS
                          ;
830:   C058 A6 B8        OPEN     LDX   LF       ;LOGISCHE FILENUMMER
840:   C05A F0 05                 BEQ   OPENERR
850:   C05C 20 0F F3              JSR   SRCHFIL   ;SUCHT FILENUMMER
860:   C05F D0 03                 BNE   *+5
870:   C061 4C FE F6 OPENERR      JMP   $F6FE    ;FILE OPEN ERROR
880:   C064 A6 98                 LDX   NMBFILES   ;ANZAHL DER OFFENEN FILES
890:   C066 E0 0A                 CPX   #10
900:   C068 90 03                 BCC   *+5
910:   C06A 4C FB F6              JMP   $F6FB    ;TOO MANY FILES ERROR
920:   C06D E6 98                 INC   NMBFILES
930:   C06F A5 B8                 LDA   LF
940:   C071 9D 59 02              STA   LFTAB,X
950:   C074 A5 B9                 LDA   SA
```

```
C076 09 60          ORA  #$60
C078 9D 6D 02       STA  SATAB,X
C07B A5 BA          LDA  FA
C07D 9D 63 02       STA  FATAB,X
C080 C9 04          CMP  #4
C082 D0 02          BNE  *+4
C084 18             CLC
C085 60             RTS            ;FERTIG
C086 C9 00          CMP  #0
C088 4C 77 F3       JMP  $F377     ;WEITER WIE GEHABT
C08B
C08B 20 14 F3 CLOSE JSR  $F314     ;SUCHT LOGISCHE FILENUMMER
C08E F0 02          BEQ  *+4
C090 18             CLC
C091 60             RTS            ;FERTIG
C092 20 1F F3       JSR  $F31F     ;SETZT FILE PARAMETER
C095 8A             TXA
C096 48             PHA
C097 A5 BA          LDA  FA
C099 C9 04          CMP  #4
C09B F0 03          BEQ  *+5
C09D 4C 9D F2       JMP  $F29D     ;NORMAL WEITER
C0A0 4C F1 F2       JMP  $F2F1     ;EINTRAG IN TABELLE LOESCHEN
C0A3 20 0F F3 CHKIN JSR  SRCHFIL    ;SUCHT FILENUMMER
C0A6 F0 03          BEQ  *+5
C0A8 4C 01 F7       JMP  $F701     ;FILE NOT OPEN ERROR
C0AB 20 1F F3       JSR  $F31F     ;SETZT FILEPARAMETER
C0AE A5 BA          LDA  FA
C0B0 C9 04          CMP  #4
C0B2 D0 03          BNE  *+5
C0B4 4C 0A F7       JMP  $F70A     ;NOT INPUT FILE ERROR
C0B7 4C 19 F2       JMP  $F219
C0BA 20 0F F3 CHKOUT JSR SRCHFIL    ;SUCHT FILENUMMER
C0BD F0 03          BEQ  *+5
C0BF 4C 01 F7       JMP  $F701     ;FILE NOT OPEN ERROR
C0C2 20 1F F3       JSR  $F31F     ;SETZT FILEPARAMETER
C0C5 A5 BA          LDA  FA
C0C7 C9 04          CMP  #4
C0C9 D0 03          BNE  *+5
C0CB 4C 75 F2       JMP  $F275
C0CE 4C 5B F2       JMP  $F25B
C0D1 48       BSOUT PHA
C0D2 A5 9A          LDA  $9A       ;OUTPUTGERAET
C0D4 C9 04          CMP  #4
C0D6 F0 03          BEQ  *+5
C0D8 4C CD F1       JMP  $F1CD
C0DB A5 B9          LDA  SA        ;SEKUNDAERADRESSE
C0DD 29 0F          AND  #%00001111
C0DF C9 00          CMP  #0
C0E1 D0 17          BNE  SEK1
C0E3 A9 3C    STAND LDA  #60       ;MODE 0 60 ZEICHEN
C0E5 8D 43 C1       STA  MAXZ
C0E8 A9 4B          LDA  #"K"      ;MODUS 480 SPROSSEN
C0EA 8D 44 C1       STA  MODE
C0ED A9 00          LDA  #0        ;NICHT BREIT
C0EF 8D 45 C1       STA  BIG
C0F2 A9 08          LDA  #8        ;SPROSSEN PRO ZEICHEN
C0F4 8D 46 C1       STA  SPZ
```

```
C0F7 4C B6 C1            JMP  OUT
C0FA C9 01      SEK1     CMP  #1
C0FC D0 17               BNE  SEK2
C0FE A9 78               LDA  #120      ;MODE 1 120 ZEICHEN
C100 8D 43 C1            STA  MAXZ
C103 A9 4C               LDA  #"L"      ;MODUS 960 SPROSSEN
C105 8D 44 C1            STA  MODE
C108 A9 00               LDA  #0
C10A 8D 45 C1            STA  BIG
C10D A9 08               LDA  #8
C10F 8D 46 C1            STA  SPZ
C112 4C B6 C1            JMP  OUT
C115 C9 02      SEK2     CMP  #2
C117 D0 17               BNE  SEK3
C119 A9 3C               LDA  #60
C11B 8D 43 C1            STA  MAXZ
C11E A9 4C               LDA  #"L"
C120 8D 44 C1            STA  MODE
C123 A9 01               LDA  #1
C125 8D 45 C1            STA  BIG
C128 A9 10               LDA  #16
C12A 8D 46 C1            STA  SPZ
C12D 4C B6 C1            JMP  OUT
C130 4C E3 C0   SEK3     JMP  STAND     ;STANDARTSCHRIFT

C133 00 00 00   MATR     .BYTE0,0,0,0,0,0,0,0  ;MATRIX-ZWISCHENSPEICHER
C13B 00         KONF     .BYTE0
C13C 00 00      CURS     .BYTE0,0       ;CURSOR X,Y MERKEN
C13E 00         ZZAHL    .BYTE0         ;ANZAHL DER ZEICHEN
C13F 00 00      ZSPAL    .BYTE0,0       ;ANZAHL DER SPROSSEN
C141 00         TEMP     .BYTE0
C142 00         SCHIRM   .BYTE0         ;ZWISCHENSPEICHER
C143 00         MAXZ     .BYTE0         ;ZEICHEN PRO ZEILE
C144 00         MODE     .BYTE0         ;MODE 480/960 SPROSSEN
C145 00         BIG      .BYTE0         ;0=NORMAL 1=BREIT
C146 00         SPZ      .BYTE0         ;SPROSSEN PRO ZEICHEN
                ;
                ;BILDSCHIRMCODE IN A WANDELN
                ;UND ZUM DRUCKER SCHICKEN
                ;
C147 8D 41 C1   CODE     STA  TEMP      ;ZEICHEN IN X
C14A 48                  PHA
C14B 98                  TYA
C14C 48                  PHA
C14D 8A                  TXA
C14E 48                  PHA
C14F AE 41 C1            LDX  TEMP
C152 A9 00               LDA  #<TABLE
C154 85 F7               STA  TABA
C156 AD 18 D0            LDA  53248+24
C159 29 02               AND  #2        ;MODUS GRAFIK ODER KLEINS.
C15B 18                  CLC
C15C 0A                  ASL
C15D 0A                  ASL
C15E 09 D0               ORA  #>TABLE
C160 85 F8               STA  TABA+1
C162 E0 00               CPX  #0        ;ADRESSE BERECHNEN
C164 F0 10               BEQ  NMAL
```

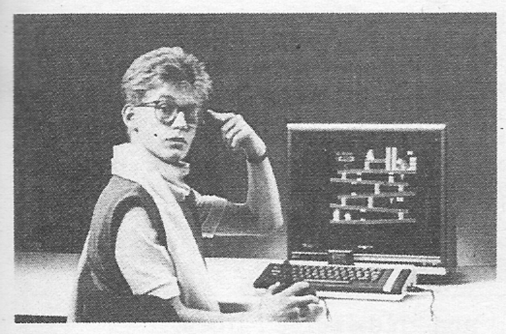
```
2120:  C166 18        MAL    CLC
2130:  C167 A5 F7            LDA  TABA
2140:  C169 69 08            ADC  #8
2150:  C16B 85 F7            STA  TABA
2160:  C16D A5 F8            LDA  TABA+1
2170:  C16F 69 00            ADC  #0
2180:  C171 85 F8            STA  TABA+1
2190:  C173 CA               DEX
2200:  C174 D0 F0            BNE  MAL
2210:  C176 A5 01     NMAL   LDA  1        ;SPEICHERKONF. MERKEN
2220:  C178 8D 3B C1         STA  KONF
2230:  C17B A9 31            LDA  #$31
2240:  C17D 78               SEI           ;KEIN IRQ
2250:  C17E 85 01            STA  1        ;NEUE KONFIGURATION
2260:  C180 A0 00            LDY  #0       ;MATRIX QUER-LAENGS WANDLUNG
2270:  C182 A2 07     UMWA   LDX  #7
2280:  C184 B1 F7            LDA  (TABA),Y
2290:  C186 6A        UMW    ROR           ;ROTIERE IN C FLAG
2300:  C187 3E 33 C1         ROL  MATR,X   ;ROTIERE C IN BYTES
2310:  C18A CA               DEX
2320:  C18B 10 F9            BPL  UMW
2330:  C18D C8               INY
2340:  C18E C0 08            CPY  #8
2350:  C190 D0 F0            BNE  UMWA
2360:  C192 AD 3B C1         LDA  KONF     ;ALTE KONF. HERSTELLEN
2370:  C195 85 01            STA  1
2380:  C197 58               CLI
2390:  C198 A2 00            LDX  #0       ;8 BYTE ZUM DRUCKER
2400:  C19A BD 33 C1  MOUT   LDA  MATR,X
2410:  C19D 48               PHA
2420:  C19E 20 40 C0         JSR  AUSGABE
2430:  C1A1 68               PLA
2440:  C1A2 AC 45 C1         LDY  BIG      ;2 MAL DRUCKEN ↑
2450:  C1A5 F0 03            BEQ  NBIG
2460:  C1A7 20 40 C0         JSR  AUSGABE
2470:  C1AA E8        NBIG   INX
2480:  C1AB E0 08            CPX  #8
2490:  C1AD D0 EB            BNE  MOUT
2500:  C1AF 68        RTS1   PLA
2510:  C1B0 AA               TAX
2520:  C1B1 68               PLA
2530:  C1B2 A8               TAY
2540:  C1B3 68               PLA
2550:  C1B4 18               CLC
2560:  C1B5 60               RTS
2570:  C1B6 68        OUT    PLA
2580:  C1B7 8D 41 C1         STA  TEMP
2590:  C1BA 48               PHA
2600:  C1BB 98               TYA
2610:  C1BC 48               PHA
2620:  C1BD 8A               TXA
2630:  C1BE 48               PHA
2640:  C1BF AD 41 C1         LDA  TEMP
2650:  C1C2 C9 0D            CMP  #$0D     ;ZEILENENDE (CR)
2660:  C1C4 D0 54            BNE  ASBI
2670:  C1C6 A9 00     CR     LDA  #0
2680:  C1C8 8D 3F C1         STA  ZSPAL
2690:  C1CB 8D 40 C1         STA  ZSPAL+1
```

```
C1CE AE 3E C1          LDX  ZZAHL    ;ANZAHL DER ZEICHEN
C1D1 F0 3A             BEQ  NMAL1    ;NUR ZEILE VORSCHIEBEN
C1D3 18       MAL1     CLC           ;ANZAHL DER SPROSSEN
C1D4 AD 3F C1          LDA  ZSPAL    ;BERECHNEN
C1D7 6D 46 C1          ADC  SPZ
C1DA 8D 3F C1          STA  ZSPAL
C1DD AD 40 C1          LDA  ZSPAL+1
C1E0 69 00             ADC  #0
C1E2 8D 40 C1          STA  ZSPAL+1
C1E5 CA                DEX
C1E6 D0 EB             BNE  MAL1
C1E8 A9 1B             LDA  #27      ;ESC ZUM DRUCKER
C1EA 20 40 C0          JSR  AUSGABE
C1ED AD 44 C1          LDA  MODE     ;EINZELPUNKT 960 SPROSSEN
C1F0 20 40 C0          JSR  AUSGABE
C1F3 AD 3F C1          LDA  ZSPAL    ;SPOSSENANZAHL DEM
C1F6 20 40 C0          JSR  AUSGABE  ;DRUCKER MITTEILEN
C1F9 AD 40 C1          LDA  ZSPAL+1
C1FC 20 40 C0          JSR  AUSGABE
C1FF A2 00             LDX  #0       ;PUFFER AUSDRUCKEN
C201 BD 3C 03 PRINT    LDA  PUFFER,X
C204 20 47 C1          JSR  CODE
C207 E8                INX
C208 EC 3E C1          CPX  ZZAHL
C20B D0 F4             BNE  PRINT
C20D A9 0D    NMAL1    LDA  #$0D     ;ZEILENVORSCHUB
C20F 20 40 C0          JSR  AUSGABE
C212 A9 00             LDA  #0
C214 8D 3E C1          STA  ZZAHL
C217 4C AF C1          JMP  RTS1
C21A AD 00 04 ASBI     LDA  1024     ;ASCII IN SCHIRMCODE
C21D 8D 42 C1          STA  SCHIRM
C220 A5 D6             LDA  $D6      ;CURSOR-POSITION MERKEN
C222 8D 3C C1          STA  CURS     ;ZEILE
C225 A5 D3             LDA  $D3
C227 8D 3D C1          STA  CURS+1   ;SPALTE
C22A 20 66 E5          JSR  $E566    ;HOME
C22D AD 41 C1          LDA  TEMP     ;ZEICHEN HOLEN
C230 20 16 E7          JSR  $E716    ;ZEICHEN AUF SCHIRM
C233 AD 00 04          LDA  1024     ;ZEICHEN VOM SCHIRM
C236 AE 3E C1          LDX  ZZAHL
C239 9D 3C 03          STA  PUFFER,X
C23C AD 42 C1          LDA  SCHIRM
C23F 8D 00 04          STA  1024
C242 AE 3C C1          LDX  CURS     ;CURSOR WIEDER SETZEN
C245 AC 3D C1          LDY  CURS+1
C248 18                CLC
C249 20 0A E5          JSR  $E50A    ;CURSOR SETZEN
C24C AE 3E C1          LDX  ZZAHL
C24F E8                INX
C250 8E 3E C1          STX  ZZAHL
C253 EC 43 C1          CPX  MAXZ
C256 D0 03             BNE  RTS2     ;ALLE 80 ZEICHEN DRUCKEN
C258 4C C6 C1          JMP  CR
C25B 4C AF C1 RTS2     JMP  RTS1
```

Computercamp
Ferienzentrum Schloß Dankern

Die Kurse

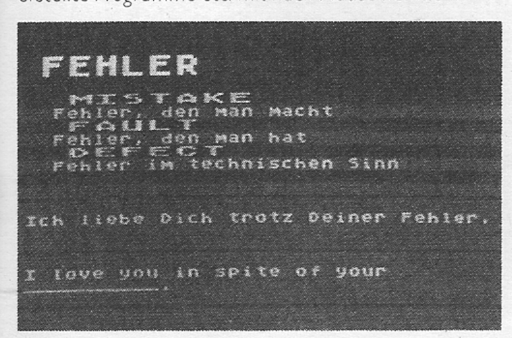
Einsteiger, Fortgeschrittene und „Cracks" werden bei uns von qualifizierten Pädagogen bzw. Informatikern betreut, die es verstehen, **individuell** auf den Wissensstand jedes Teilnehmers einzugehen und Informationen **spielerisch** zu vermitteln.



Die angebotene Palette umfaßt:

- Einführung in Hardware und -Bedienung
- Einführung in die Kommunikation mit dem Computer
- Einführung in die Programmiersprachen
- Die Programmiersprache BASIC von A–Z
- BASIC für Fortgeschrittene
- Perfektionieren von Programmen in BASIC
- Einführung und Programmieren in Maschinensprache

Darüberhinaus aber natürlich **jede Menge detaillierter Information** wie zum Beispiel BASIC-Dialekte, oder wie man einen bereits vorhandenen Computer optimal nutzen kann, etc.
Da der Erfolg der Kurse auch von der Verfügbarkeit der Geräte abhängt, **garantieren wir jedem Teilnehmer ein eigenes Gerät,** das er mindestens **3 Stunden täglich** nutzen kann.
Und – wie gesagt – für individuelle Betreuung ist genauso gesorgt wie für Raum zur Entfaltung von Kreativität und Eigeninitiative – das fördert den wichtigen Erfahrungsaustausch.
Zum Abschluß des Camps erhält jeder Teilnehmer ein **Abschlußzertifikat,** und kann natürlich auch selbst erstellte Programme etc. mit nach Hause nehmen.



Wir freuen uns schon auf Sie – bis bald!

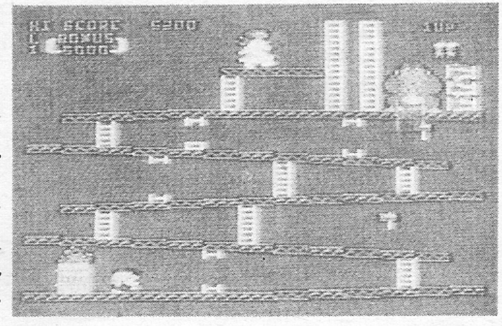
Die Freizeitmöglichkeiten

Fast zuviel, um alles aufzuzählen: **Ponyreiten, Schwimmbad, Minigolf, Tischtennis, Riesenspielplatz** mit **Tarzanschaukel, Westernfort, Fahrradkarussel, Pferderennen, Kletterturm, Riesenrutsche... alles im Preis enthalten!** Darüberhinaus **Kegelbahnen, Autoscooter, Modellflugplatz, Fernsehräume,** und auf dem Wasser **Wasserskilift, Windsurfen, Wasserfahrräder, Tret- und Ruderboote...**

Nicht zu vergessen das 300 Jahre alte **Schloß,** das zur Besichtigung einlädt, genauso wie mehrere **Cafés, Restaurants,** ein **Ferienkino,** eine **Super-Disco...**

Haben wir etwas vergessen? Ach ja – einkaufen kann man auch – sogar sonntags!

Wir finden: Ein insgesamt überzeugendes Angebot, das sicher auch Computerfans begeistern wird.



Die Termine

Die Camps beginnen jeweils an einem Samstag. Die Belegung ist möglich für 1 Woche, 14 Tage oder 3 Wochen (Pfingsten nur 4 Tage).

Ostern '84
14.4.–21.4.
21.4.–28.4.

Pfingsten '84
[redacted]
9.6.–12.6.

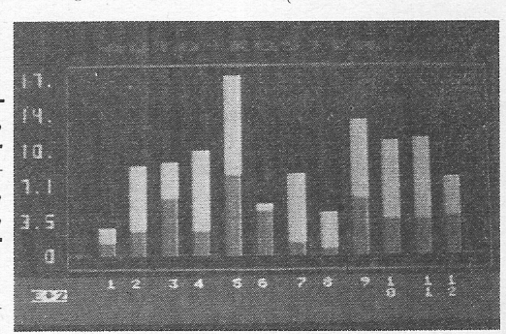
Sommer '84
23.6.–30.6.
30.6.– 7.7.
7.7.–14.7.
14.7.–21.7.
21.7.–28.7.
28.7.– 4.8.
4.8.–11.8.
11.8.–18.8.
18.8.–25.8.
25.8.– 1.9.
1.9.– 8.9.

Herbst '84
6.10.–13.10.
13.10.–20.10.
20.10.–27.10.
27.10.– 3.11.

Die Kosten

4 Tage (nur Pfingsten)	350,– DM
1 Woche	550,– DM
2 Wochen	980,– DM
3 Wochen	1480,– DM

Die Preise gelten für Unterbringung **mit Vollpension; die Kurse und fast alle Freizeitmöglichkeiten sind inklusive.**
Sondertarife für Gruppen oder Schulklassen auf Anfrage. Eltern, die ihre Kinder begleiten wollen, sind dazu herzlich eingeladen – rufen Sie uns einfach an!



Die Unterbringung

Gewohnt wird nicht in Hotels, Jugendherbergen oder Zeltlagern, sondern in **supergemütlichen Ferienhäusern** (4–6 Personen) aus Holz, die in unmittelbarer Nähe des Schlosses Dankern und des dazugehörigen Dankern-Sees liegen, inmitten eines **150 Hektar großen Ferienzentrums.** Übrigens können Eltern auch gerne ihre Kinder begleiten – Anfrage beim Buchungsbüro genügt.

Frühstück, Mittag- und Abendessen sind inklusive.





Die Anreise

Ort der Handlung ist das **Ferienzentrum Schloß Dankern,** 4472 Haren (Ems). Ein riesiges, landschaftlich wunderschön gelegenes Feriengebiet mitten im schönen Emsland. Wie man hinkommt, ist am besten der Karte zu entnehmen. Die Bahn (Bahnhof Haren) ist nur etwa 2 km entfernt. Weitere Fragen beantworten wir gerne – rufen Sie uns bitte an.

Buchungsbüro Computercamp
Ferienzentrum Schloß Dankern
Holztwiete 4 D
2000 Hamburg 52
Tel.: (040) 82 79 42



– tödliche Strahlen greifen an

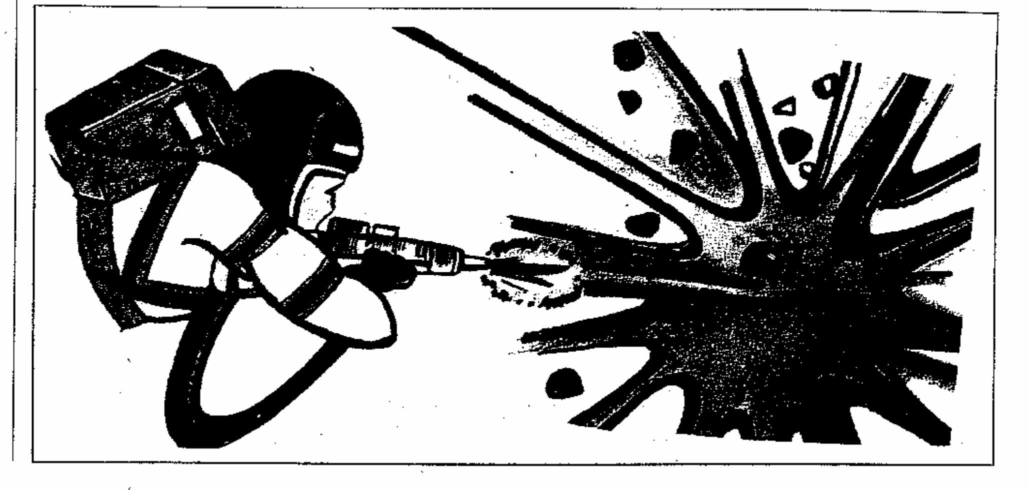
Über Ihrem Planeten ziehen farbige Wolken auf. Diese Wolken schicken totbringende Strahlen zur Erde. Wenn einer dieser Strahlen eine der sechs Städte erreicht, so wird die Stadt zerstört und somit sämtliches Leben vernichtet.
Es liegt nun an Ihnen diese Invasion zu vereiteln.
Nehmen Sie Platz in Ihrer LASER 2001 Komandozentrale und zerstören Sie die tödlichen Strahlen, bevor diese sämtliche Städte vernichtet haben.
Das Spiel wird mit dem linken Joystick gesteuert. Um auch mit einem ATARI Joystick spielen zu können, wurde der linke Feuerknopf zum Abfeuern der Laserkanone belegt.
Tippen Sie das Listing ein und speichern Sie dieses mit CSAVE auf Cassette ab. Starten Sie jetzt das Programm mit RUN. Sie werden als erstes nach einer Spielanleitung gefragt. Bei Eingabe von „J" erhalten Sie eine kurze Anleitung auf dem Bildschirm. Danach können Sie eine von 10 Spielversionen wählen. Dies hat den Vorteil, daß Sie anfangs mit einer sehr leichten Version trainieren können, später jedoch gleich bei einer schwierigeren Variante beginnen können.
Sie haben die Wahl zwischen folgenden 10 Versionen:
1 Kinderversion
2 Blutiger Anfänger
3 Anfänger
4 leichte Version
5 Normalversion
6 Mittelklasse
7 Fortgeschrittene
8 Könner
9 Profi
10 Vollprofi
Die Unterschiede in den verschiedenen Versionen können Sie Tabelle 2 entnehmen.
Wenn Sie sich für eine bestimmte Version entschieden haben, so geben Sie bitte die Nummer der entsprechenden Version ein und drücken die -RETURN- Taste.
Jetzt werden Sie aufgefordert den Taster am linken Joystick zu drücken, um zu kontrollieren, ob Sie den richtigen Joystick an die richtige Buchse angeschlossen haben. Wenn das geschehen ist, erscheint das Titelbild: Die Laserkanone erscheint, die Titelschrift wird ausgegeben und das Lied vom Tod liegt in der Luft.
Nachdem das Lied vom Tod verklungen ist, wird über Ihrer Laserkanone ein kleiner weißer Zielpunkt sichtbar. Diesen Punkt können Sie mit dem Joystick in 8 verschiedene Richtungen bewegen. Der Punkt kann jedoch nur in einem bestimmten Bildschirmbereich bewegt werden, und gibt den Zielpunkt Ihres Laserstrahles an.
Die Strahlen kommen aus den farbigen Wolken und zielen immer auf eine existierende Stadt. Sie können einen Strahl vernichten, indem Sie diesen an seiner Spitze treffen. Dazu steuern Sie den weißen Zielpunkt in die Nähe der Spitze des Strahles und drücken den Feuerknopf. Wenn der Zielpunkt nicht weiter als 10 Bildschirmpunkte von der Strahlspritze entfernt ist, so wird der Todesstrahl zerstört. Feuern Sie jedoch einen Laserstrahl ab, welcher den Todesstrahl nicht trifft, so werden Ihnen im Energieband (grünes Band am unteren Bildschirmrand mit „E" gekennzeichnet) eine Energieeinheit abgezogen. Wenn Sie in diesem Energieband keine Einheiten mehr zur Verfügung haben, so schießt der Laser nicht mehr, und Sie müssen tatenlos zusehen, wie Ihre Städte zerstört werden.
Die Todesstrahlen treten in Wellen auf. Wenn Sie eine Welle gemeistert haben, so werden die geretteten Städte abgerechnet. Danach folgt die nächste, schnellere Welle an Todesstrahlen.
Die Wertung gliedert sich wie folgt:

Welle	Geschwindigkeit	Zerstörung Todesstrahl	Anzahl der Strahlen	gerettete Stadt
1	1 fach	10 Punkte	8	100 Punkte
2	2 fach	20 Punkte	10	200 Punkte
3	3 fach	30 Punkte	12	300 Punkte
4	4 fach	40 Punkte	14	400 Punkte
5	5 fach	50 Punkte	16	500 Punkte
6	6 fach	60 Punkte	18	600 Punkte
7	7 fach	70 Punkte	20	700 Punkte
8	8 fach	80 Punkte	22	800 Punkte
9	9 fach	90 Punkte	24	900 Punkte
10	10 fach	100 Punkte	26	1000 Punkte

Das Spiel ist beendet, wenn alle sechs Städte zerstört sind. Es erscheint dann GAME OVER und das Lied vom Tod ertönt.
Wenn das Lied vom Tod verklungen ist, so können Sie ein neues Spiel durch drücken einer beliebigen Taste auf der Konsole beginnen.
Tabelle 2: Unterschiede der Varianten

Variante	Beginn bei Welle	Einheiten im Energieband
1	1	160
2	2	152
3	3	144
4	4	136
5	5	128
6	6	120
7	7	112
8	8	104
9	9	96
10	10	88

Die Geschwindigkeit steigert sich bis zur Welle 10. Danach geht das Spiel zwar weiter, erhöht die Geschwindigkeit jedoch nicht mehr. Es werden aber dann nach 26 Strahlen weiterhin die Städte abgerechnet.
Wenn die Städte abgerechnet werden, ändern diese die Farben.
Tips zum Eintippen:
Verwenden Sie soviel als möglich die -CTRL- Tasten für die Eingabe. Das erspart Tippfehler.
Anstelle von PRINT kann jedesmal ? eingegeben werden. Im Listing steht dann nachher wieder PRINT.
Sämtliche Leerzeichen in den BASIC Anweisungen können weggelassen werden. Sie werden vom Rechner automatisch beim Listing generiert.
Schreiben Sie das Programm vor dem ersten Start mit RUN mehrmals weg (CSAVE), da der LASER durch die hohe Schreibgeschwindigkeit sonst das Programm nicht mehr, oder nicht mehr korrekt lesen könnte.
Wenn das Spiel mittendrin einfach stehen bleibt, so ist ein Tippfehler vorhanden. Gehen Sie dann mit -CTRL- -S- in den TEXT Modus um die Fehlermeldung zu sehen.
Nach dem ersten Start muß sich bei der Abfrage ?RAM(0)-16384 ein Wert in der Größenordnung 2100 bis 2310, je nach Abbruchstelle im Programm ergeben.



```
1   REM  ***************************************
2   REM  *           A N D R O M E D A          *
3   REM  *--------------------------------------*
4   REM  *  fuer LASER 2001 und linken Joystick  *
5   REM  *--------------------------------------*
6   REM  *        (c) 1984 Volker Becker         *
7   REM  *            Steinbacher Strasse 10     *
8   REM  *            6370 Oberursel 6           *
9   REM  ***************************************
10  REM
11  REM
80  SOUND (255,1,0),(255,1,0),(255,1,0),(1,1,0,1)
90  CLEAR
100  DIM SF(6,2)
110  FOR I = 1 TO 6:SF(I,1) = 1: NEXT I
120 SF(1,2) = 24:SF(2,2) = 56:SF(3,2) = 88:SF(4,2) = 167:SF(5,2) = 199:SF(6,2) =
 231
130  TEXT
135  HOME
136  PRINT
140  COLOR= 1,2
145  PRINT  CHR$ (24); CHR$ (24); CHR$ (24); SPC( 9);"A N D R O M E D A"
150  PRINT  SPC( 9);"-----------------"; CHR$ (24)
155  PRINT "   toetliche Strahlen greifen an"
160  FOR I = 1 TO 16: PRINT  CHR$ (24);: NEXT I
165  PRINT  SPC( 5);"(c) 1984 by Volker Becker"; CHR$ (5)
170  FOR I = 1 TO 14: PRINT  CHR$ (5);: NEXT I
175  PRINT  SPC( 6);"Spielanleitung ? (J/N)";
180  FOR I = 15 TO 0 STEP  - 1: SOUND (180,1,I),(190,1,I),(200,1,I): NEXT I
185  SOUND (255,1,0),(255,1,0),(255,1,0),(1,1,0,1)
190  GET S$: IF S$ = "J" THEN  GOTO 35000
195  IF S$ = "N" THEN  GOTO 200
197  GOTO 190
200  HOME
205  PRINT
210  PRINT "    S P I E L V E R S I O N E N"
215  PRINT "    ---------------------------"
220  PRINT  CHR$ (24); CHR$ (24); CHR$ (24); SPC( 7);"1  Kinderversion"
225  PRINT  SPC( 7);"2  Blutiger Anfaenger"
230  PRINT  SPC( 7);"3  Anfaenger"
235  PRINT  SPC( 7);"4  leichte Version"
240  PRINT  SPC( 7);"5  Normalversion"
245  PRINT  SPC( 7);"6  Mittelklasse"
250  PRINT  SPC( 7);"7  Fortgeschrittene"
260  PRINT  SPC( 7);"8  Koenner"
265  PRINT  SPC( 7);"9  Profi"
270  PRINT  SPC( 6);"10  Vollprofi"; CHR$ (24); CHR$ (24); CHR$ (24); CHR$ (24)
275  PRINT  CHR$ (5); SPC( 7);"Ihre Wahl bitte : ";
280  FOR I = 15 TO 0 STEP  - 1: SOUND (180,1,I),(190,1,I),(200,1,I): NEXT I
285  SOUND (255,1,0),(255,1,0),(255,1,0),(1,1,0,1)
290  INPUT RU
300 RU =  INT (RU)
310  IF RU < 1 OR RU > 10 THEN  GOTO 275
320  PRINT  CHR$ (24); CHR$ (24);" Taster am linken Joysick druecken"
330  CALL  - 1536
335  IF  PEEK (21) = 17 THEN  GOTO 950
340  GOTO 330
950  HOME
960  GR
962  GOSUB 50200
963  COLOR= 8,1
965  FOR I = 175 TO 191: PLOT 0,I TO 255,I
970  NEXT I
```

```
980 P = 0:EG = 168 - (RU * 8)
990  GOSUB 50500: GOSUB 31000
995  GOSUB 42000
1002 C = 2
1003  GOSUB 42000
1004  FOR V = 1 TO 2
1005  FOR S = 5 TO 250 STEP 8:Z = 5 +  INT ( RND (1) * 6): GOSUB 50400
1006 C = C + 1: IF C > 14 THEN C = 2
1007  NEXT S
1008  NEXT V: GOSUB 60000: GOSUB 49900
1009 RU = RU - 1
1010 Z = 161:S = 17: GOSUB 50000: SOUND (10,2,15)
1020 S = 49: GOSUB 50000: SOUND (20,2,15)
1030 S = 81: GOSUB 50000: SOUND (30,2,15)
1040 S = 160: GOSUB 50000: SOUND (40,2,15)
1050 S = 192: GOSUB 50000: SOUND (50,2,15)
1060 S = 224: GOSUB 50000: SOUND (60,2,15)
1070 RU = RU + 1
1100  FOR AN = 1 TO 6 + 2 * RU
2000 A =  INT ( RND (255) * 5 + 1.5): IF SF(A,1) = 0 THEN  GOTO 2000
2010 B =  INT ( RND (SF(A,2)) * 234 + 10)
2015 B1 = B
2020 ST = SF(A,2) - B
2030 ST = (ST / 146) * RU
2035 SV = 127:ZV = 140
2040 C = 1
2050  FOR Z = 15 TO 160 STEP RU
2052  FOR I = 1 TO 3
2053  CALL  - 1536
2054 JS =  PEEK (24)
2055  IF JS <  > 0 THEN  GOTO 10000
2056  COLOR= 15,1: PLOT SV,ZV
2057  NEXT I
2060  COLOR= C,1:C = C + 1: IF C > 15 THEN C = 2
2070  PLOT B,15 TO B1,Z
2080 B1 = B1 + ST
2090  NEXT Z
2100  GOSUB 41000
2105  UNPLOT SV,ZV
2106  FOR VG = 1 TO 6: IF SF(VG,1) = 1 THEN  GOTO 2120
2107  NEXT VG
2108  GOTO 50900
2120  NEXT AN
2140  FOR SA = 1 TO 6: IF SF(SA,1) = 1 THEN S = SF(SA,2) - 7:Z = 161: GOSUB 5000
0: SOUND (10 * SA,2,15):P = P + 100 * RU: GOSUB 31000
2150  NEXT SA
2155 RU = RU + 1: IF RU > 10 THEN RU = 10
2160  GOTO 1100
10000 JS = JS / 2
10005  IF JS > 50 THEN  GOTO 20000
10010  IF JS > 10 THEN  GOTO 11000
10020  ON JS GOTO 10100,10200,10300,10400,256,10600
10100  UNPLOT SV,ZV
10110  IF ZV >  = 140 THEN  GOTO 2056
10120 ZV = ZV + 4
10130  GOTO 2056
10200  UNPLOT SV,ZV
10210  IF SV >  = 245 THEN  GOTO 2056
10220 SV = SV + 4
10230  GOTO 2056
10300  UNPLOT SV,ZV
10310  IF ZV >  = 140 OR SV >  = 245 THEN  GOTO 2056
10320 SV = SV + 4:ZV = ZV + 4
```

```
10330  GOTO 2056
10400  UNPLOT SV,ZV
10410  IF ZV <  = 50 THEN  GOTO 2056
10420 ZV = ZV - 4
10430  GOTO 2056
10600  UNPLOT SV,ZV
10610  IF ZV <  = 50 OR SV >  = 245 THEN  GOTO 2056
10620 ZV = ZV - 4:SV = SV + 4
10630  GOTO 2056
10700  UNPLOT SV,ZV
10710  IF SV <  = 10 THEN  GOTO 2056
10720 SV = SV - 4
10730  GOTO 2056
10800  UNPLOT SV,ZV
10810  IF ZV >  = 140 OR SV <  = 10 THEN  GOTO 2056
10820 SV = SV - 4:ZV = ZV + 4
10830  GOTO 2056
10900  UNPLOT SV,ZV
10910  IF ZV <  = 50 OR SV <  = 10 THEN  GOTO 2056
10920 SV = SV - 4:ZV = ZV - 4
10930  GOTO 2056
11000 JS = JS - 15
11010  ON JS GOTO 10700,10800,2056,2056,10900
20000  COLOR= 13,1
20005  IF EG = 0 THEN  GOTO 2056
20010  PLOT 127,159 TO SV,ZV
20012  SGEN 190,10,180,220,210,255
20013  SGEN 1,1,2,2,3,19
20014  FOR LG = 1 TO 50: NEXT LG
20015  SGEN 159,191,223,255
20016  UNPLOT 127,159 TO SV,ZV
20017  IF  ABS (ZV - (Z - RU)) <  = 10 THEN  GOTO 20030
20018 EG = EG - 1: COLOR= 2,1: UNPLOT EG + 16,184 TO EG + 16,186
20020  GOTO 2056
20030  IF (B1 - SV) <  = 10 THEN  GOTO 21000
20040  GOTO 20018
21000 S = SV:Z = ZV: GOSUB 30000
21010  GOTO 2105
30000  SGEN 159,191,223,255
30005  SGEN 100,77,210,110
30010 SD = 1
30015  FOR I = 1 TO 10
30020  SGEN SD,SD + 32,SD + 10,SD + 64,SD + 5,SD + 48
30030  COLOR= C,1:C = C + 1: IF C > 15 THEN C = 2
30040  CIRCLE (S,Z),I
30050  NEXT I
30070  FOR K =  - 10 TO 10
30080  COLOR= C,1:C = C + 1: IF C > 15 THEN C = 15
30090  SGEN SD,SD + 32,SD + 10,SD + 64,SD + 5,SD + 48
30100 SD = SD + 1
30110  UNPLOT S - 10,Z - K TO S + 10,Z + K
30114  UNPLOT B + K,15 TO B1 + K,Z - RU
30115  UNPLOT B + K,15 TO B1 + K,Z
30116  UNPLOT B + K,15 TO B1 + K,Z + RU
30120  NEXT K
30130  SOUND (255,1,0),(255,1,0),(255,1,0),(1,1,0;1)
30140  SGEN 159,191,223,255
30150  SGEN 100,77,210,110
30170  FOR K =  - 10 TO 10
30180  COLOR= C,1:C = C + 1: IF C > 15 THEN C = 15
30190  SGEN SD,SD + 32,SD + 10,SD + 64,SD + 5,SD + 48
30200 SD = SD + 1
30210  UNPLOT S - K,Z + 10 TO S + K,Z - 10
```

```
30220  NEXT K
30226  SGEN 159,191,223,255
30227 P = P + 10 * RU: GOSUB 31000
30230  RETURN
31000  COLOR= 8,1
31005 P$ =  STR$ (P)
31007 S = 240 - 7 *  LEN (P$)
31010 L =  VAL ( MID$ (P$,1,1)) + 1
31020  PLOT S + 1,181 TO S + 5,181 TO S + 5,189 TO S + 1,189 TO S + 1,181
31025  PLOT S + 2,185 TO S + 4,185
31030  ON L GOSUB 40000,40100,40200,40300,40400,40500,40600,40700,40800,40900
31035  IF  LEN (P$) <  = 1 THEN  RETURN
31040 P$ =  RIGHT$ (P$, LEN (P$) - 1)
31050  GOTO 31007
35000  HOME
35005  COLOR= 1,7
35010  PRINT
35020  PRINT  SPC( 9);"A N D R O M E D A"
35030  PRINT  SPC( 9);"-----------------"
35040  PRINT
35050  PRINT "Toetliche Strahlen aus dem Weltall  greifen die Erde an."
35060  PRINT "Ihre Aufgabe ist es nun diese        Strahlen zu zerstoeren, bevor
diese Ihre sechs Staedte vernichtet haben"
35070  PRINT "Bewegen Sie mit dem Joystick den     weissen Zielpunkt an die gewue
nschteStelle, und druecken Sie den linken Aktionsknopf."
35080  PRINT "Wenn Sie die Spitze des Strahles im Umkreis von 10 Bildpunkten mit
 dem  Laser erreicht haben, so wird der   Strahl vernichtet."
35090  PRINT "Gelingt es jedoch einem Strahl bis  zu einer Stadt vorzudringen, s
o      wird diese zerstoert."
35100  PRINT "Das Spiel ist beendet, wenn alle    sechs Staedte vernichtet sind.
"
35105  PRINT
35110  PRINT  SPC( 8);"Weiter mit <RETURN>"
35120  GET A$: IF A$ =  CHR$ (13) THEN  GOTO 35140
35130  GOTO 35120
35140  HOME
35150  PRINT
35160  PRINT "Achten Sie besonders auf das        Energieband, denn fuer jeden S
chuss der daneben geht, wird Ihnen eine   Energieeinheit abgezogen."
35170  PRINT "Wenn Sie keine Energie mehr haben,  so muessen Sie tatenlos zusehe
n, wieIhre Staedte zerstoert werden."
35180  PRINT "Es stehen Ihnen 10 Versionen zur    Verfuegung, die sich in der St
art-  runde und in der Laenge des Energie-bandes unterscheiden."
35190  PRINT  CHR$ (24);"Sie erhalten pro zerstoertem Strahl 10 * Runde Punkte.
Nach jeder Runde werden die geretteten Staedte mit   100 * Runde Punkte abgerech
net."
35200  PRINT "Pro Runde erscheinen 6 + 2 * Runde  Strahlen."
35210  PRINT  CHR$ (24);"  Doch nun viel Spass beim Spiel"
35220  PRINT  CHR$ (24); SPC( 8);"Weiter mit <RETURN>"
35230  GET A$: IF A$ =  CHR$ (13) THEN  GOTO 35250
35240  GOTO 35230
35250  COLOR= 1,2
35260  GOTO 200
40000  FOR I = 181 TO 189 STEP 8: UNPLOT S + 2,I TO S + 4,I
40010  NEXT I
40020  FOR K = 1 TO 5 STEP 4
40030  FOR I = 182 TO 186 STEP 4: UNPLOT S + K,I TO S + K,I + 2
40040  NEXT I
40050  NEXT K
40060  RETURN
40100  UNPLOT S + 5,182 TO S + 5,184
40110  UNPLOT S + 5,186 TO S + 5,188
40120  RETURN
```

```
40200  FOR I = 181 TO 189 STEP 4: UNPLOT S + 2,I TO S + 4,I
40210  NEXT I
40220  UNPLOT S + 5,182 TO S + 5,184
40230  UNPLOT S + 1,186 TO S + 1,188
40240  RETURN
40300  FOR I = 181 TO 189 STEP 4: UNPLOT S + 2,I TO S + 4,I
40310  NEXT I
40320  UNPLOT S + 5,182 TO S + 5,184
40330  UNPLOT S + 5,186 TO S + 5,188
40340  RETURN
40400  UNPLOT S + 1,182 TO S + 1,184
40410  UNPLOT S + 5,182 TO S + 5,184
40420  UNPLOT S + 2,185 TO S + 4,185
40430  UNPLOT S + 5,186 TO S + 5,188
40440  RETURN
40500  FOR I = 181 TO 189 STEP 4: UNPLOT S + 2,I TO S + 4,I
40510  NEXT I
40520  UNPLOT S + 1,182 TO S + 1,184
40530  UNPLOT S + 5,186 TO S + 5,188
40540  RETURN
40600  FOR I = 181 TO 189 STEP 4: UNPLOT S + 2,I TO S + 4,I
40610  NEXT I
40620  UNPLOT S + 1,182 TO S + 1,184
40630  UNPLOT S + 5,186 TO S + 5,188
40640  UNPLOT S + 1,186 TO S + 1,188
40650  RETURN
40700  UNPLOT S + 2,181 TO S + 4,181
40710  UNPLOT S + 5,182 TO S + 5,184
40720  UNPLOT S + 5,186 TO S + 5,188
40730  RETURN
40800  FOR I = 181 TO 189 STEP 4: UNPLOT S + 2,I TO S + 4,I
40810  NEXT I
40830  FOR K = 1 TO 5 STEP 4
40840  FOR I = 182 TO 186 STEP 4: UNPLOT S + K,I TO S + K,I + 2
40850  NEXT I
40860  NEXT K
40870  RETURN
40900  FOR I = 181 TO 189 STEP 4: UNPLOT S + 2,I TO S + 4,I
40910  NEXT I
40920  UNPLOT S + 5,182 TO S + 5,184
40930  UNPLOT S + 5,186 TO S + 5,188
40940  UNPLOT S + 1,182 TO S + 1,184
40950  RETURN
41000 S = SF(A,2):Z = 164
41070  FOR K =  - 10 TO 10
41080  COLOR= C,1:C = C + 1: IF C > 15 THEN C = 15
41090  SOUND (200 - K,1,15),(190 - K,1,15),(205 - K,1,15)
41110  UNPLOT S - 10,Z - K TO S + 10,Z + K
41120  NEXT K
41226  SOUND (255,1,0),(255,1,0),(255,1,0),(1,1,0,1)
41228  SGEN 100,77,210,110
41230  FOR I =  - 10 TO 10
41235 SD = I + 11
41237  SGEN SD,SD + 32,SD + 10,SD + 64,SD + 5,SD + 48
41240  UNPLOT B + I,15 TO B1 + I,161
41242  UNPLOT B1,161 TO B1 + I,174
41243  SGEN SD,SD + 32,SD + 10,SD + 64,SD + 5,SD + 48
41250  NEXT I
41253  SGEN 159,191,223,255
41255  SOUND (255,1,0),(255,1,0),(255,1,0),(1,1,0,1)
41260  FOR I = 161 TO 174: UNPLOT S - 10,I TO S + 10,I
41270  NEXT I
41275 SF(A,1) = 0
```

```
41280  RETURN
42000 S = 16:Z = 56
42005  FOR C = 2 TO 10: COLOR= C,1
42010  READ SK
42020  FOR I = 0 TO 2
42030  ON SK GOSUB 51000,52000,53000,54000,55000,56000,57000,58000,59000
42040  NEXT I
42050 S = S + 24
42060  NEXT C
42070  RETURN
49900  FOR I = 54 TO 78: UNPLOT 0,I TO 255,I
49910  NEXT I
49920  RETURN
50000  COLOR= RU + 2,1
50010  PLOT S + 7,Z TO S + 7,Z + 1
50020  FOR I = 2 TO 3: PLOT S + 6,Z + I TO S + 8,Z + I
50030  NEXT I
50040  FOR I = 4 TO 8: PLOT S + 5,Z + I TO S + 9,Z + I
50050  NEXT I
50060  FOR I = 9 TO 10: PLOT S,Z + I TO S + 9,Z + I
50070  NEXT I
50080  FOR I = 11 TO 13: PLOT S,Z + I TO S + 14,Z + I
50090  NEXT I
50100  FOR I = 12 TO 13: PLOT S + I,Z + 5 TO S + I,Z + 10
50110  NEXT I
50120  PLOT S + 2,Z + 7
50130  PLOT S + 1,Z + 8 TO S + 3,Z + 8
50140  UNPLOT S + 2,Z + 10
50150  UNPLOT S + 7,Z + 4 TO S + 7,Z + 5
50160  FOR I = 11 TO 13 STEP 2: UNPLOT S + I,Z + 12
50170  NEXT I
50180  RETURN
50200  COLOR= 11,1
50210  PLOT 127,160 TO 127,162
50220  PLOT 128,160 TO 128,162
50230  COLOR= 6,1
50240  FOR I = 163 TO 165: PLOT 126,I TO 129,I
50250  NEXT I
50260  COLOR= 5,1
50270  FOR I = 166 TO 170: PLOT 125,I TO 130,I
50280  NEXT I
50290  UNPLOT 127,166 TO 128,166
50300  UNPLOT 126,167 TO 126,168
50310  UNPLOT 129,167 TO 129,168
50320  UNPLOT 127,169 TO 128,169
50330  COLOR= 2,1
50340  FOR I = 120 TO 124: PLOT I,175 TO I + 4,171
50350  NEXT I
50360  FOR I = 135 TO 131 STEP - 1: PLOT I,175 TO I - 4,171
50370  NEXT I
50380  RETURN
50400  COLOR= C,1
50410  FOR I = 1 TO 5: CIRCLE (S,Z),I: NEXT I
50420  FOR I =  - 5 TO 5 STEP 3: UNPLOT S - 5,Z - I TO S + 5,Z + I
50430  NEXT I
50440  FOR I =  - 5 TO 5 STEP 3: UNPLOT S - I,0 TO S + I,10
50450  NEXT I
50460  RETURN
50500  COLOR= 8,1
50501  FOR I = 181 TO 189 STEP 4: UNPLOT 10,I TO 12,I
50502  NEXT I
50503  UNPLOT 9,181 TO 9,189
50510  UNPLOT 15,187 TO 16 + EG,187 TO 16 + EG,183 TO 15,183 TO 15,187
```

```
50520  COLOR= 2,1
50530  FOR I = 184 TO 186: PLOT 16,I TO 15 + EG,I
50540  NEXT I
50550  RETURN
50900  FOR K = 1 TO 8: READ S,Z,SK,C
50919  COLOR= C,1
50920  FOR I = 0 TO 2
50930  ON SK GOSUB 51000,52000,53000,54000,55000,56000,57000
50935  NEXT I
50936  NEXT K
50940  GOSUB 60000
50950  GET A$
50970  IF A$ <  > "" THEN  GOTO 50980
50975  GOTO 50950
50980  RUN
51000  PLOT S + 19 + I,Z + 1 - I TO S + 2 + I,Z + 1 - I TO S + 2 + I,Z + 22 - I
TO S + 19 + I,Z + 22 - I TO S + 19 + I,Z + 12 - I TO S + 7 + I,Z + 12 - I
51010  RETURN
52000  PLOT S + 2 + I,Z + 22 - I TO S + 10 + I,Z + 1 - I TO S + 11 + I,Z + 1 - I
 TO S + 19 + I,Z + 22 - I
52010  PLOT S + 6 + I,Z + 12 - I TO S + 15 + I,Z + 12 - I
52020  RETURN
53000  PLOT S + 2 + I,Z + 22 - I TO S + 2 + I,Z + 1 - I TO S + 10 + I,Z + 12 - I
 TO S + 11 + I,Z + 12 - I TO S + 19 + I,Z + 1 - I TO S + 19 + I,Z + 22 - I
53010  RETURN
54000  PLOT S + 19 + I,Z + 1 - I TO S + 2 + I,Z + 1 - I TO S + 2 + I,Z + 22 - I
TO S + 19 + I,Z + 22 - I
54010  PLOT S + 2 + I,Z + 12 - I TO S + 15 + I,Z + 12 - I
54020  RETURN
55000  RECT (S + 2 + I,Z + 1 - I),(S + 19 + I,Z + 22 - I)
55010  RETURN
56000  PLOT S + 2 + I,Z + 1 - I TO S + 10 + I,Z + 22 - I TO S + 11 + I,Z + 22 -
I TO S + 19 + I,Z + 1 - I
56010  RETURN
57000  PLOT S + 2 + I,Z + 22 - I TO S + 2 + I,Z + 1 - I TO S + 15 + I,Z + 1 - I
TO S + 19 + I,Z + 5 - I TO S + 19 + I,Z + 8 - I TO S + 15 + I,Z + 12 - I TO S +
6 + I,Z + 12 - I TO S + 19 + I,Z + 22 - I
57010  RETURN
58000  PLOT S + 2 + I,Z + 22 - I TO S + 2 + I,Z + 1 - I TO S + 19 + I,Z + 22 - I
 TO S + 19 + I,Z + 1 - I
58010  RETURN
59000  PLOT S + 2 + I,Z + 1 - I TO S + 15 + I,Z + 1 - I TO S + 19 + I,Z + 5 - I
TO S + 19 + I,Z + 18 - I TO S + 15 + I,Z + 22 - I TO S + 2 + I,Z + 22 - I TO S +
 2 + I,Z + 1 - I
59010  RETURN
59999  DATA  2,8,9,7,5,3,4,9,2
60000  DATA   95,255,20,160,95,50,165,148,10,160,135,10,148,128,10,135,113,50,13
5,255,20,184,165,20,210,184,40,207,175,10,201,165,10,207,175,60,201,165,20,201,1
65,20,195,160,30,192,148,10,195,160,10,184,135,10
60002  DATA   175,128,60,175,255,20,184,165,20,210,184,20,201,165,30,195,255,10,
207,175,60
60003  DATA    175,255,20,184,165,20,210,184,20,201,165,30,195,160,10,195,160,60
,0,0,0
60010  RESTORE
60015  FOR I = 1 TO 9: READ SK: NEXT I
60020  READ H1,H2,V: IF H1 = 0 THEN  GOTO 60055
60030  SOUND (H1,V,15),(H2,V,13)
60050  GOTO 60020
60055  SOUND (255,1,0),(255,1,0),(255,1,0),(1,1,0,1)
60056  RETURN
60057  SGEN 159,191,223,225
60070  DATA     40,48,1,5,88,48,2,10,136,48,3,8,184,48,4,2,40,88,5,13,88,88,6,4,
136,88,4,14,184,88,7,7
```

Block Painter



Das Spiel „Reversal" dürfte eigentlich fast jedem Computerbesitzer bekannt sein, denn es ist auf vielen Computern als Programm verfügbar. Hier ist nun die Apple-Version dieses Spiels. Die Spielregeln werden, falls noch nicht bekannt, auch vom Programm erklärt, wenn man das wünscht. Weiterhin kann man noch zwischen zwei Spielstärken wählen und dem Computer mitteilen, ob man orangefarbene oder blaue Spielsteine haben will.

Orange beginnt das Spiel. Das elektronische Spielbrett wird auf dem Bildschirm samt Bezeichnungen der Felder in der Applegrafik farbig dargestellt. Der Spieler gibt das Feld, auf das er setzen will, an, indem er jeweils einen Buchstaben für die vertikale Stellung und eine Zahl für die horizontale angibt.

Der Computer bestätigt den Zug und führt alle notwendigen Veränderungen auf dem Brett automatisch aus. Dann „denkt" er eine Weile und gibt seinen Zug bekannt.

Falsche Benutzereingaben werden abgefangen. Man sollte daran denken, daß man immer nur einen Buchstaben und eine Zahl eingibt, ohne <RET> zu drücken.

Sind alle Felder belegt oder hat ein Spieler keine Steine mehr, dann rechnet der Computer den Gewinner aus und gibt eventuell noch einen Kommentar zum Spielverlauf.

Das Spiel kann auch zu zweit gespielt werden, wobei der Bildschirm eben nur als Spielbrett dient. Man muß dann am Anfang eingeben, daß man nicht gegen den Apple spielen will. Ob man hier nicht vielleicht ein reales Spielbrett vorzieht, bleibt jedem selbst überlassen.

Bei der Eingabe des Programms muß man beachten, daß vor allem die ersten Zeilen bis zur maximalen Zeilenlänge vollgepackt sind. Man sollte beim Eintippen alle Freiräume weglassen, sonst verweigert der Apple die Annahme der Zeile.

Die Spielstärke ist für die „Denkzeit" ganz annehmbar und der Anfänger hat bestimmt eine Weile zu üben, bis er zum ersten Mal gewinnt. Viel Spaß dabei!

```
1  LOMEM: 24870: DIM A(9,9),B(7),C(7),B$(2): FOR C = 0 TO 7: READ B(C),C(
    C): NEXT : DATA 0,1,-1,1,-1,0,-1,-1,0,-1,1,-1,1,0,1,1: GOSUB 93: SCALE=
    1: ROT= 0: POKE 232,0: POKE 233,96:N$ = "12345678":L$ = "
                                      "
2  TEXT : HOME : PRINT "GREETINGS FROM REVERSI": PRINT "ANLEITUNG BENOETI
    GT ? ";: GET C$: IF C$ = "N" THEN  PRINT "NEIN": GOTO 8
3  PRINT "JA": PRINT : HOME : COLOR= 3: HLIN 0,39 AT 0: VTAB 2: PRINT "#R
    EVERSI IS PLAYED ON AN 8 * 8 BOARD,  #";: PRINT "#ROWS NUMBERED 1 - 8
     AND COLUMNS A - H #";: PRINT "#THE INITIAL CONFIGURATION IS ALL BLAN
    K#";: PRINT "#EXCEPT FOR THE CENTER FOUR SQUARES    #";: PRINT "#TRY
    TO PLACE YOUR PIECE SO THAT IT    #";
4  PRINT "#OUTFLANKS MINE, CREATING A HORIZONTAL,#";: PRINT "#VERTICAL,OR
     DIAGONAL RUN OF MY PIECES #";: PRINT "#TURNING THEM INTO YOURS
           #";: PRINT "#                                  #";: PRINT
    "#MAKE YOUR MOVE BY ENTERING A NUMBER   #";
5  PRINT "#FOR THE ROW AND A LETTER FOR THE      #";: PRINT "#COLUMN.
                             ##                                     #"
    ;: PRINT "#NOTE: YOU MUST CAPTURE AT LEAST ONE OF##POSSIBLE. IF IT IS
     NOT POSSIBLE, YOU  #";
6  PRINT "#FORFEIT YOUR MOVE BY ENTERING 0 A FOR ##YOUR MOVE
                  #";: PRINT "#                                    ##
    IF YOU UNDERSTAND, PRESS A KEY         #";
7  PRINT "#                                 ##
                  ########################################";: VTAB 20:
     HTAB 33: GET A$
8 D(0) = 5:D$(0) = "ORANGE":D(2) = 6:D$(2) = "BLAU":D = 2:E = D:F = 4:H =
    0:K = 1:L = 0:M = 0:N = 0: HOME : PRINT : PRINT "WILLST DU GEGEN DEN
    APPLE SPIELEN->";: GET C$: IF C$ = "N" THEN  PRINT "NEIN": GOTO 13
9  PRINT " JA":K = 3: PRINT : PRINT "SOLL ICH SO GUT SPIELEN,": PRINT "WI
    E ICH KANN -->";: GET C$: IF C$ = "N" THEN  PRINT " NEIN": GOTO 11
10  PRINT " JA":L = 2:M = 1:N =  - 2
11  PRINT : PRINT "ORANGE FAENGT IMMER AN ...": PRINT : PRINT "WILLST DU
    ORANGE ODER BLAU --> ";: GET C$: IF C$ = "O" THEN  PRINT D$(0): GOTO
    13
12  PRINT D$(2):K = 2
13 B =  - 1:O = 1: FOR I = 0 TO 9: FOR J = 0 TO 9:A(I,J) = 0: NEXT J,I:A(
    4,4) = O:A(5,5) = O:A(4,5) = B:A(5,4) = B: GOSUB 92: IF K = 2 THEN  VTAB
    21: PRINT "OK - ICH DENKE                    "
14 P = B:Q = O: ON K GOSUB 57,70,57
15  IF H = 1 THEN 17
16 D = D + R + 1:E = E - R:F = F + 1: GOSUB 51
17  IF E = 0 OR F = 64 OR H < 0 THEN 23
```



```
18 P = O:Q = B: ON K GOSUB 57,57,70
19  IF H = 1 THEN 21
20 E = E + R + 1:D = D - R:F = F + 1: GOSUB 51
21  IF D = 0 OR F = 64 OR H < 0 THEN 23
22  GOTO 14
23  HOME : VTAB 21: PRINT "ORANGE HAT "D" UND BLAU HAT "E" STEINE.": INVERSE
    : IF BC = WC THEN 26
24  IF D > E THEN 27
25  PRINT "BLAU";: GOTO 28
26  PRINT "UNENDSCHIEDEN !!": NORMAL : GOTO 39
27  PRINT "ORANGE";
28  PRINT "  GEWINNT !":D = D - E: NORMAL : IF D > 0 THEN 30
29 D =  - D
30 D = (64 * D) / F: PRINT "DAS WAR EIN ";: IF D < 11 THEN 38
31  IF D < 25 THEN 37
32  IF D < 39 THEN 36
33  IF D < 53 THEN 35
34  PRINT "PERFEKTES SPIEL": GOTO 39
35  PRINT "DURCHMARSCH": GOTO 39
36  PRINT "KAMPF": GOTO 39
37  PRINT "HEISSES SPIEL !": GOTO 39
38  PRINT "WAHNSINN !": GOTO 39
39  PRINT "WILLST DU NOCHMAL SPIELEN ";: GET C$: IF C$ = "J" THEN  PRINT
    "JA": RUN 7
40  PRINT "NEIN": TEXT : HOME : PRINT "DANKE FUERS SPIELEN !!": END
41 S = 0: FOR T =  - 1 TO 1: FOR V =  - 1 TO 1: IF A(I + T,J + V) = Q THEN
    43
42  NEXT V,T: RETURN
43 S = 1: RETURN
44 R = 0: FOR C = 0 TO 7:W = B(C):X = C(C):Y = I + W:Z = J + X:A1 = 0: IF
    A(Y,Z) <  > Q THEN 50
45 A1 = A1 + 1:Y = Y + W:Z = Z + X: IF A(Y,Z) = P THEN 48
46  IF A(Y,Z) = 0 THEN 50
47  GOTO 45
48 R = R + A1: IF U <  > 1 THEN 50
49 Y = I:Z = J: FOR C1 = 0 TO A1:A(Y,Z) = P:Y = Y + W:Z = Z + X: NEXT
50  NEXT : RETURN
51  FOR I = 1 TO 8: FOR J = 1 TO 8: IF A(I,J) <  > 0 THEN  HCOLOR= D(A(I,
    J) + 1): DRAW 1 AT J * 30,I * 17 + 1
52  NEXT : NEXT : RETURN
53 I =  - 1:J = I: FOR C = 1 TO 2: GET C$:G =  ASC (C$): IF 47 < G AND G <
    58 THEN I = G - 48: GOTO 55
54  IF 64 < G AND G < 74 THEN J = G - 64: GOTO 55
55  PRINT C$" ";: NEXT : PRINT "": RETURN
56  VTAB 24: INVERSE : PRINT  CHR$ (7);"FALSCHER ZUG ! NUR 0-8 UND A-H ":
     NORMAL : FOR D1 = 1 TO 3000: NEXT : VTAB 23: PRINT E$
57  VTAB 21: PRINT "EINGABE "D$(P + 1)"'S ZUG -->       ";: POKE 36, PEEK
    (36) - 5: GOSUB 53: IF I < 0 OR J < 0 OR I > 8 OR J > 8 THEN 56
58  IF I <  > 0 THEN 62
59  VTAB 24: PRINT "LAESST "D$(P + 1)" DIESEN ZUG AUS ?";: GET C$: IF C$ =
    "N" THEN  PRINT "NEIN": FOR D1 = 1 TO 2000: NEXT : VTAB 23: PRINT E$:
     GOTO 57
60  PRINT "JA": FOR D1 = 1 TO 2000: NEXT : VTAB 23: PRINT E$: IF H = 1 THEN
    H =  - H: RETURN
61 H = 1: RETURN
62  IF A(I,J) = 0 THEN 64
```



```
63  VTAB 24: INVERSE : PRINT  CHR$ (7);"FELD SCHON VON "D$(A(I,J) + 1)" B
    ESETZT": NORMAL : FOR D1 = 1 TO 3000: NEXT : VTAB 23: PRINT E$: GOTO
    68
64  GOSUB 41: IF S = 1 THEN 66
65  VTAB 24: INVERSE : PRINT  CHR$ (7);"IST NICHT AN "D$(Q + 1)" DRAN ": NORMAL
    : FOR D1 = 1 TO 3000: NEXT : VTAB 23: PRINT E$: GOTO 68
66 U =  - 1: GOSUB 44: IF R > 0 THEN 69
67  VTAB 24: INVERSE : PRINT  CHR$ (7);"SCHLEAGT "D$(Q + 1)" NICHT": NORMAL
    : FOR D1 = 1 TO 3000: NEXT : VTAB 23: PRINT E$
68  GOTO 57
69  HOME : VTAB 22: GOTO 89
70 B1 =  - 1:E1 = 0:F1 = 0: FOR I = 1 TO 8: FOR J = 1 TO 8: IF A(I,J) <  >
    0 THEN 83
71  GOSUB 41: IF S = 0 THEN 83
72 U =  - 1: GOSUB 44: IF R = 0 THEN 83
73  IF I = 1 OR I = 8 THEN R = R + L
74  IF J = 1 OR J = 8 THEN R = R + L
75  IF I = 2 OR I = 7 THEN R = R + N
76  IF J = 2 OR J = 7 THEN R = R + N
77  IF I = 3 OR I = 6 THEN R = R + M
78  IF J = 3 OR J = 6 THEN R = R + M
79  IF R < B1 THEN 83
80  IF R > B1 THEN 82
81  IF  RND (1) > .5 THEN 83
82 B1 = R:E1 = I:F1 = J
83  NEXT J,I: IF B1 > 0 THEN 88
84  IF N = 0 THEN 86
85 N = 0: GOTO 70
86  VTAB 24: INVERSE : PRINT D$(P + 1)" LAESST DIESEN ZUG AUS ": FOR D1 =
    1 TO 4000: NEXT : NORMAL : VTAB 23: PRINT E$: IF H = 1 THEN H =  - H:
     RETURN
87 H = 1: RETURN
88 I = E1:J = F1: HOME : VTAB 22
89  PRINT D$(P + 1)" SETZT NACH " CHR$ (I + 48)"  " CHR$ (J + 64)"
       ":H = 0:U = 1: GOSUB 44: VTAB 23: PRINT "DAS GIBT ";D$(P + 1);" ";
    R;: IF R > 1 THEN  PRINT " STEINE  ": GOTO 91
90  PRINT " STEIN   "
91  RETURN
92  HGR : HCOLOR= 1: FOR I = 15 TO 159 STEP 17: HPLOT 20,I TO 260,I: NEXT
    : FOR I = 21 TO 261 STEP 30: HPLOT I,15 TO I,151: NEXT : GOSUB 51: HCOLOR=
    3: FOR I = 1 TO 8: DRAW I + 1 AT 7,I * 17 + 5: NEXT : FOR I = 1 TO 8:
     DRAW I + 9 AT I * 30 + 3,0: NEXT : RETURN
93  FOR I = 24576 TO 24865: READ A: POKE I,A: NEXT : RETURN
94  DATA 18,0,38,0,91,0,99,0,109,0,120,0,131,0,144,0,156,0,164,0,178,0,19
    5,0,211,0,223,0,237,0,250,0,5,1,17,1,109,58,73,9,109,58,63,63,23,45,4
    5,45,173,63,63,63,63,55,45,45,45,45,173,63
95  DATA 63,63,63,63,55,45,45,45,45,45,245,63,63,63,63,55,45,45,45,45,245
    ,63,63,63,119,45,45,245,59,7,0,9,62,14,54,190,45,5,0,98,45,21,246,191
    ,23,46,45,45,0,45,45,30,23,21,21,246,63,7,32,0
96  DATA 73,62,23,13,254,51,45,45,30,54,0,45,45,222,27,46,45,21,54,30,63,
    7,32,0,9,245,187,46,45,21,246,63,7,32,4,0,45,45,246,23,30,54,6,0,41,1
    73,223,51,77,241,63,23,118,45,5,32,4,0,10,45
97  DATA 21,223,51,77,49,223,51,45,45,254,27,110,137,48,0,42,45,21,223,51
    ,77,241,63,55,77,49,223,51,45,45,0,10,45,21,223,51,54,110,9,30,63,7,0
    ,42,173,223,110,9,254,27,110,9,30,223,46,45,0,42,45,245
98  DATA  219,54,45,245,27,54,45,45,5,0,42,45,245,219,54,45,245,27,54,6,0
    ,10,45,245,219,54,54,14,45,37,36,63,0,106,9,254,27,110,9,62,63,55,77,
    49,223,51,77,17,48,0
```

```
99  :
100 :
101 REM ***********************
102 REM *                     *
103 REM *                     *
104 REM *       REVERSI       *
105 REM * (C) ROBERT TOLKSDORF *
106 REM *     AUGASSE 25      *
107 REM * 875 ASCHAFFENBURG   *
108 REM *                     *
109 REM *                     *
110 REM *                     *
111 REM *  LENGTH : 5.3 KBYTE *
112 REM *                     *
113 REM ***********************
114 :
115 :
```



Disk-Menü-Generator

Dieses Applesoft-Programm erstellt für jede beliebige Diskette ein Menü-Programm, das auf Knopfdruck beliebige Programme startet. Nach dem erstellen des Menü-Programmes, kann man dieses mit „SAVE NAME" auf die Diskette bringen. Der Name sollte so gewählt werden, das das Programm nach PR#6 immer gebootet wird.

Start des Generators:

1. RUN MENÜ GENERATOR
2. Die entsprechende Disk einlegen und Taste drücken
3. Editiermodus (siehe unten)
4. Menü Programm ist im Speicher und kann mit SAVE... abgespeichert werden.

Editiermodus:
Es erscheint der erste Filename auf der Diskette. Soll dieser im Menü gleich bleiben so bestätigen Sie RETURN Soll der Name im Menü anders lauten so geben Sie die neue Bezeichnung ein und drücken RETURN. Soll das Programm nicht im Menü aufgeführt werden, z. B. bei Unterprogrammen, Grafiken usw., so geben Sie einfach das folgende Zeichen ein: „-" RETURN. Nun kommt der nächste Filename und das ganze wiederholt sich.

Der Menü-Generator achtet darauf das Applesoftprogramme mit „RUN" Maschinenprogramme mit „BRUN" und Text files mit „EXEC" gestartet werden.

SO SIEHT DAS MENÜ AUF DEM SCHIRM AUS: Beispiel eines Listing, das vom Programm erstellt wurde.

Das Listing des oben abgedruckten Menü's sieht wie folgt aus:

```
1   DIM M$(100)
10   HOME
20   PRINT "****** DISK-MENUE-GENERATOR ******": PRINT : PRINT : PRINT
23   FLASH : PRINT " DISKETTE EINLEGEN !!! ";
100  INP$ = ""
110  VS =  PEEK (106) * 256 +  PEEK (105)
120   POKE VS + 2,240: POKE VS + 3,14: POKE VS + 4,150
125   GET W$: IF W$ = "" THEN 125
126   NORMAL :W$ = "": HOME
130   REM LOAD RWTS CONTROL ROUTINE & IOB AT $300 TO $319
140   DATA 169,3,160,9,32,217,3,96,0,1,96,1,0,17,15,251,183,0,150,0,0,1,0,2
      54,96,1
150   FOR I = 768 TO 793
160   READ X: POKE I,X
170   NEXT I
180  BL$ = ""
190   FOR I = 1 TO 40
200  BL$ = BL$ +  CHR$ (160): NEXT I
210   DIM N$(105),S$(105),T$(105),L$(105)
230  D$ =  CHR$ (4):B$ =  CHR$ (7)
240  IOB = 777:T = 64:PL = 15
250  RB =  - 27136: REM READ BUFFER
260  CS = IOB + 1:CD = IOB + 2:TRK = IOB + 4:SEC = IOB + 5
270  VOL = IOB + 14:OS = IOB + 15:OD = IOB + 16:F1 = RB + 11:F7 = RB + 221
320  S =  PEEK ( - 18455) / 16:D =  PEEK ( - 18454):K = 0
330   POKE S,S * 16: POKE CD,D
340   POKE TRK,17
350  F = 496:P = 1:C = 0
360   TEXT : PRINT " ### DISK-MENUE-GENERATOR ###": PRINT
490   FOR X = 15 TO 1 STEP  - 1
500   POKE SEC,X: CALL 768
510  ZZ =  - 34
520   FOR Y = F1 TO F7 STEP 35
530  ZZ = ZZ + 35
540  Z =  PEEK (Y + 33): IF Z = 0 THEN 690
550  C = C + 1
560   IF  PEEK (Y) = 255 THEN N$(C) = "1": GOTO 670
570  F = F - Z
580  S$(C) =  STR$ (Z)
590   IF  LEN (S$(C)) = 1 THEN S$(C) = "00" + S$(C)
600   IF  LEN (S$(C)) = 2 THEN S$(C) = "0" + S$(C)
610  Z =  PEEK (Y + 2):L$(C) = " ": IF Z > 16 THEN L$(C) = "*":Z = Z - 128
620   IF Z = 0 THEN T$(C) = "T"
630   IF Z = 1 THEN T$(C) = "I"
640   IF Z = 2 THEN T$(C) = "A"
650   IF Z = 4 THEN T$(C) = "B"
```

```
655  IF Z = 16 THEN T$(C) = "R"
660 N$(C) =  MID$ (INP$,ZZ,30)
670  NEXT Y
680  NEXT X
690 L% = C / PL: IF C > L% * PL THEN L% = L% + 1
700 V$ =  STR$ ( PEEK (VOL))
710  IF  LEN (V$) = 1 THEN V$ = "00" + V$
720  IF  LEN (V$) = 2 THEN V$ = "0" + V$
730 F$ =  STR$ (F): IF  LEN (F$) = 1 THEN F$ = "00" + F$
740  IF  LEN (F$) = 2 THEN F$ = "0" + F$
750 : HOME : PRINT "VOLUME NR."V$"  FREIE SEKTOREN:"F$: PRINT
760  FOR I = 1 TO PL
770  IF (J * PL + I) > C THEN 810
780  IF N$(J * PL + I) = "1" THEN  PRINT  CHR$ (T + I);"                  ";: INVERSE
     : PRINT "DELETED FILE": NORMAL
790 Z = 1
793  IF N$(Z) = "" THEN 800
795 Z = Z + 1: GOTO 793
800  REM
810 Q = 10:D$ =  CHR$ (4)
820  FOR I = 1 TO Z - 1
822  IF N$(I) = "1" THEN 860
830  VTAB 5: PRINT "                                            ": PRINT "
                                "
840  VTAB 5: PRINT "FILE NAME : "N$(I)
845  HTAB 13: PRINT N$(I);: HTAB 1: VTAB 7
850  INPUT "MENUE NAME: ";M$(I)
855  IF M$(I) = "" THEN M$(I) = N$(I)
860  NEXT I
870  HOME
871  FLASH : PRINT : PRINT : PRINT : PRINT "DAS MENUE-PROGRAMM WIRD ERSTEL
     LT !"
872  PRINT : PRINT : PRINT : HTAB 15: PRINT "AUGENBLICK": NORMAL
900  PRINT D$"OPEN MENUE.FB"
910  PRINT D$"WRITE MENUE.FB"
915  PRINT "NEW"
920  PRINT "1 HOME:?"; CHR$ (34);"DISK MENUE 1.0      FREIE SEKTOREN:"; CHR$
     (34);F$;":?"
929 U = 0
930  FOR I = 1 TO Z - 1
931 U = U + 1
933  IF N$(I) = "1" THEN U = U - 1: GOTO 990
934  IF M$(I) = "-" THEN U = U - 1: GOTO 990
935  PRINT Q"?" CHR$ (34); CHR$ (64 + U);" -- "M$(I); CHR$ (34)
980 Q = Q + 10
990  NEXT I
1000  PRINT Q"?:?"; CHR$ (34);"BITTE WAEHLEN : "; CHR$ (34);";"
1010 Q = Q + 10
1020  PRINT Q"GET E$:IFE$=" CHR$ (34); CHR$ (34);"THEN"Q
1024 J = Q
1025 Q = Q + 10: PRINT Q"?"
1030 Q = Q + 10
1039 U = 0
1040  FOR I = 1 TO Z - 1
1041 U = U + 1
1045  IF N$(I) = "1" THEN U = U - 1: GOTO 1090
1046  IF M$(I) = "-" THEN U = U - 1: GOTO 1090
1050  IF T$(I) = "A" THEN B$ = "RUN "
1055  IF T$(I) = "B" THEN B$ = "BRUN "
1060  IF T$(I) = "I" THEN B$ = "RUN "
1070  IF T$(I) = "T" THEN B$ = "EXEC "
```



```
1080  PRINT Q"IF E$=" CHR$ (34); CHR$ (64 + U); CHR$ (34);"THEN ?C
                    HR$(4);" CHR$ (34);B$;N$(I); CHR$ (34)
1090 Q = Q + 1: NEXT I
1100 Q = Q + 9: PRINT Q"RUN"
1110  PRINT Q + 1"REM"
1120  PRINT Q + 2"REM ***** MENUE 1.0 *****"
1130  PRINT Q + 3"REM **  COMPUTRONIC    **"
1135  PRINT "HOME:VTAB6:?" CHR$ (34)"MENUEPROGRAMM IST IM SPEICHER !" CHR$
     (34)
1140  PRINT "DELETE MENUE.FB"
2000  PRINT D$"CLOSE MENUE.FB"
2100  PRINT D$"EXEC MENUE.FB"
```

```
                    DISK MENUE 1.0      FREIE SEKTOREN:129

                    A -- --
                    B -- ADVENTURE LAND
                    C -- VOODOO CASTLE
                    D -- MATHEMATIK I
                    E -- ZOMBIE-ISLAND
                    F -- RIBBIT
                    G -- TELEFON

                    BITTE WAEHLEN :
```

```
1  HOME : PRINT "DISK MENUE 1.0      FREIE SEKTOREN:"129: PRINT

10  PRINT "A -- --"
20  PRINT "B -- ADVENTURE LAND                   "
30  PRINT "C -- VOODOO CASTLE                    "
40  PRINT "D -- MATHEMATIK I                     "
50  PRINT "E -- ZOMBIE-ISLAND                    "
60  PRINT "F -- RIBBIT                           "
70  PRINT "G -- TELEFON                          "
80  PRINT : PRINT "BITTE WAEHLEN : ";
90  GET E$: IF E$ = "" THEN 90
100  PRINT
110  IF E$ = "A" THEN  PRINT  CHR$ (4);"RUN HELLO
                    "
111  IF E$ = "B" THEN  PRINT  CHR$ (4);"BRUN ADVENTURE LAND
                     "
112  IF E$ = "C" THEN  PRINT  CHR$ (4);"BRUN VOODOO CASTLE
                     "
113  IF E$ = "D" THEN  PRINT  CHR$ (4);"RUN MATHEMATIK I
                    "
114  IF E$ = "E" THEN  PRINT  CHR$ (4);"RUN ZOMBIE-ISLAND
                    "
115  IF E$ = "F" THEN  PRINT  CHR$ (4);"BRUN RIBBIT
                     "
116  IF E$ = "G" THEN  PRINT  CHR$ (4);"RUN TELEFON
                    "
126  RUN
127  REM
128  REM  ***** MENUE 1.0 *****
129  REM  **  COMPUTRONIC    **
```

Solitaire



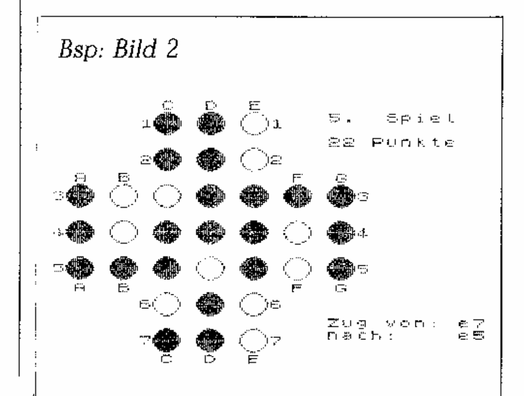
Ihre Aufgabe in diesem Spiel ist es, durch Springen mit den blauen Steinen, ein verzwicktes Problem zu lösen.
Es darf jeweils nur ein Stein übersprungen werden, und auch nur dann, wenn hinter diesem Stein ein weißes Loch frei ist. Außerdem dürfen Sie nur senkrecht oder waagerecht über den Stein springen. Sprünge über Eck oder diagonal sind nicht erlaubt. Die übersprungenen Steine werden entfernt. Können Sie nicht mehr springen, dann werden die restlichen Steine als Fehlerpunkte angerechnet.
Und jetzt zum Problem:
Sie müssen so geschickt springen, daß erstens nur ein Stein übrigbleibt (1 Punkt) und zweitens dieser Stein am Schluß genau im mittleren Loch des Spielfeldes sitzt (0 Punkte).
Damit Sie einen besseren Überblick über den Spielablauf bekommen, wird Sie der Computer durch das erste Spiel führen.
Und nun viel Spaß!

EINGABEN
Protokoll:
Über die Taste >P< können Sie sich jederzeit sämtliche Züge anzeigen oder ausdrucken lassen, auch während eines Spiels.
Neues Spiel:
Wenn Sie die Taste >N< drücken, beginnt ein neues Spiel für Sie.
Ende:
Drücken Sie die Taste >E<, wird der Programmablauf abgebrochen.
Auto:
Wenn Sie die Taste >A< drücken, übernimmt der Computer das nächste Spiel.
Copy:
Drücken Sie während eines Spiels die beiden Tasten >CO< dann wird der momentane Bildschirminhalt ausgedruckt.
Leistungskurve:
Nach jedem Spiel haben Sie die Möglichkeit, durch Drücken der Taste >L< Ihre Gesamtleistung in einem Schaubild zu betrachten.

EINGABE
Geben Sie zuerst das kurze Programm (Zeile 1 und 2) ein. Speichern Sie das Programm mit GOTO2 auf einer Kassette ab. Nachdem Sie jetzt das Programm gelöscht haben, geben Sie das zweite Programm ein. Starten Sie das Programm und warten Sie, bis in der unteren Hälfte des Bildschirms das Wort >ENTER< erscheint. Brechen Sie dann das Programm wieder ab. Das zweite Programm sollte jetzt unmittelbar hinter dem ersten Programmteil mit GOTO 4000 abgespeichert werden.

LADEN
Laden Sie das erste Programm mit LOAD'' oder LOAD 'Solitaire'. Das zweite Programm wird jetzt automatisch geladen und gestartet.

Bsp: Bild 2




```
PROTOKOLL
---------
 1. Zug: d2d4        12. Zug: e3c3
 2. Zug: f3d3        13. Zug: d4b4
 3. Zug: d4d2        14. Zug: c1e1
 4. Zug: b4d4        15. Zug: e4e6
 5. Zug: b3d3        16. Zug: g3e3
 6. Zug: f5f3        17. Zug: e6c6
 7. Zug: e1e3        18. Zug: c7e7
 8. Zug: c6c4        19. Zug: b5d5
 9. Zug: e5c5        20. Zug: e2e4
10. Zug: e7e5        21. Zug: g5g3
11. Zug: c2e2        22. Zug: a4c4
                     23. Zug: c4c2
```

```
Leistungskurve
--------------
P  0
U  2
N  4
K  6
T  8
E 10
  12
  14
  16
  18
  20
  22
  24
  26
  28
   1 3 5 7 9 1 1 1 1 1 2 2 2
             1 3 5 7 9 1 3 5
ENTER                   SPIEL
```

Wichtig:

```
USR-DEFINED-GRAPHICS
--------------------
A =          B =
C =          D =
E =          F =
G =          H =
I =          J =
K =          L =
M =          N =
O =          P =
Q =          R =
S =          T =
U =
```

```
   1 BORDER 1: PAPER 1: INK 6: CLS : LOAD "Solitaire"SCREEN$ : PRINT AT 17,0;: MERGE "Solitaire" : GO TO 10
   2 SAVE "Solitaire" LINE 1
  10 DIM i(50,4): DIM h(25): LET le=1: LET au=0: GO SUB 8800: GO TO 4100
  15 LET au=0
  20 DIM a(11,11): DIM d$(31,4): DIM b$(1,2): DIM c$(1,4): LET en=0
 100 GO SUB 8660: LET zu=0: BORDER 1: PAPER 1: INK 6: CLS : PRINT PAPER 5; INK 1;AT 0,6;"     "
 120 FOR z=1 TO 19 STEP 3: PRINT PAPER 5; INK 1;AT z,6;"      ";AT z+1,6;"      ";AT z+2,6;"      ": NEXT z
 135 PRINT PAPER 5; INK 1;AT 6,0;"      ";AT 6,16;"      "
 150 FOR z=7 TO 13 STEP 3: PRINT PAPER 5; INK 1;AT z,0;"      ";AT z,16;"      ";AT z+1,0;"      ";AT z+1,16;"      ";AT z+2,0;"      ";AT z+2,16;"      ": NEXT
```

```
z
 170 GO SUB 6000: PRINT PAPER 5; INK 1;AT 10,10;"⌒";AT 11,10;"◡": GO SUB 6020: LET l=USR 40000
 190 PRINT PAPER 5; INK 1;AT 1,6;"ᵢ";AT 1,15;"ᵢ";AT 2,6;"⁺";AT 2,15;"⁺";AT 4,6;"₂";AT 4,15;"₋";AT 5,6;"ᴸ";AT 5,15;"ᴱ"
 200 PRINT PAPER 5; INK 1;AT 7,0;"₃";AT 7,21;"₃";AT 8,0;"⌣";AT 8,21;"⌣";AT 10,0;"₄";AT 10,21;"₄";AT 11,0;"⁺";AT 11,21;"⁺";AT 13,0;"₌";AT 13,21;"₌";AT 14,0;"⌣";AT 14,21;"⌣"
 210 PRINT PAPER 5; INK 1;AT 16,6;"₅";AT 16,15;"₅";AT 17,6;"⌣";AT 17,15;"⌣";AT 19,6;"₇";AT 19,15;"₇";AT 20,6;"ʳ";AT 20,15;"ʳ"
 212 GO SUB 6000: LET pu=32: PRINT AT 3,19;pu;" Punkte": PRINT AT 1,19;le;".";TAB 23;"Spiel"
 215 IF au=1 THEN GO SUB 2500: GO TO 218
 217 IF en=0 THEN INPUT "Zug von:";b$(1);
 218 LET c$(1,1)=b$(1,1): LET c$(1,2)=b$(1,2)
 222 IF en=1 THEN INPUT INK 1; PAPER 6;"P"; INK 6; PAPER 1;"rotokoll   "; INK 1; PAPER 6;"N"; INK 6; PAPER 1;"eues Spiel   "; INK 1; PAPER 6;"E"; INK 6; PAPER 1;"nde   "; INK 1; PAPER 6;"L"; INK 6; PAPER 1;"eistungskurve   "; INK 1; PAPER 6;"A"; INK 6; PAPER 1;"uto:  ";b$(1)
 223 GO SUB 6100: IF b$(1,1)="p" THEN GO SUB 700: GO SUB 1000: GO TO 217
 224 IF en=1 AND (b$(1,1)="n" OR b$(1,1)="a") AND le>25 THEN GO SUB 710: GO TO 222
 225 IF en=1 AND b$(1,1)="n" THEN GO SUB 700: GO TO 15
 226 IF en=1 AND b$(1,1)="l" THEN GO SUB 700: GO SUB 2000: GO TO 222
 227 IF en=1 AND b$(1,1)="e" THEN GO SUB 700: STOP
 228 IF en=1 AND b$(1,1)="a" THEN GO SUB 700: LET au=1: GO TO 20
 229 IF en=1 THEN GO SUB 700: GO TO 222
 230 LET e1=0: LET d1=0: LET d2=0: LET k1=CODE b$(1,1)-96: LET k2=CODE b$(1,2)-48
 290 IF k1<1 OR k1>7 OR k2<1 OR k2>7 THEN GO SUB 710: GO TO 215
 300 IF a(k2+2,k1+2)=2 OR a(k2+2,k1+2)=0 THEN GO SUB 710: GO TO 215
 310 PRINT AT 18,19;"Zug von:  ";b$(1);AT 19,19;"          ": GO SUB 700
 320 IF au=1 THEN GO SUB 2600: GO TO 327
 325 INPUT "nach:";b$(1);
 327 LET c$(1,3)=b$(1,1): LET c$(1,4)=b$(1,2)
 330 LET n1=CODE b$(1,1)-96: LET n2=CODE b$(1,2)-48
 350 IF n1<1 OR n1>7 OR n2<1 OR n2>7 THEN GO SUB 710: GO TO 215
 360 IF a(n2+2,n1+2)=1 OR a(n2+2,n1+2)=0 THEN GO SUB 710: GO TO 215
 365 IF n2-k2=0 THEN LET d1=ABS (n1-k1)
 370 IF n1-k1=0 THEN LET d2=ABS (n2-k2)
 375 IF d1=2 THEN LET e1=e1+1
 380 IF d2=2 THEN LET e1=e1+1
 385 IF e1<>1 THEN GO SUB 710: GO TO 215
 390 LET s1=n1-k1: LET s2=n2-k2
 400 IF s1<0 OR s2<0 THEN LET t1=-1
 405 IF s1>0 OR s2>0 THEN LET t1=1
 420 IF k1=n1 THEN LET v1=k2+t1: IF a(v1+2,k1+2)<>1 THEN GO SUB 710: GO TO 215
 422 IF k1=n1 THEN LET a(v1+2,k1+2)=2: PRINT PAPER 5; INK 1;AT (v1-1)*3+1,(k1-1)*3+1;"⌒";AT (v1-1)*3+2,(k1-1)*3+1;"◡"
 425 IF k2=n2 THEN LET v1=k1+t1: IF a(k2+2,v1+2)<>1 THEN GO SUB 710: GO TO 215
 430 IF k2=n2 THEN LET a(k2+2,v1+2)=2: PRINT PAPER 5; INK 1;AT (k2-1)*3+1,(v1-1)*3+1;"⌒";AT (k2-1)*3+2,(v1-1)*3+1;"◡"
 500 PRINT AT 19,19;"nach:     ";b$(1): GO SUB 700: LET a(k2+2,k1+2)=2: LET a(n2+2,n1+2)=1
 522 PRINT PAPER 5; INK 1;AT (k2-1)*3+1,(k1-1)*3+1;"⌒";AT (k2-1)*3+2,(k1-1)*3+1;"◡"
 523 PRINT PAPER 5; INK 1;AT (n2-1)*3+1,(n1-1)*3+1;"▄▄";AT (n2-1)*3+2,(n1-1)*3+1;"▀▀"
 532 LET pu=pu-1: PRINT AT 3,19;"  ";AT 3,19;pu
 535 IF pu=1 THEN PRINT AT 3,22;"Punkt "
 540 LET zu=zu+1: LET d$(zu)=c$(1): GO SUB 1500: GO TO 215
 700 BEEP .1,10: RETURN
 710 BEEP .1,20: BEEP .1,20: RETURN
1000 LET l=USR 50000: POKE 50001,0: POKE 50002,64: POKE 50007,40: POKE 50008,160
1020 CLS : PRINT "Protokoll";AT 1,0;"---------": INPUT "ausdrucken? (J/N)  ";f$: GO SUB 700
1027 IF f$="j" THEN LPRINT "PROTOKOLL": LPRINT "---------"
1030 FOR z=1 TO zu
1040 IF z<=20 THEN PRINT z;".";TAB 4;"Zug:  ";d$(z): GO TO 1046
1045 PRINT AT z-19,15;"|   ";z;".";TAB 22;"Zug:  ";d$(z)
1046 IF f$="j" THEN LPRINT z;".";TAB 4;"Zug:  ";d$(z)
1048 NEXT z
1049 PRINT AT 21,21; PAPER 6; INK 1; FLASH 1;"ENTER";AT 18,21;""
1050 LET b$(1)="": PAUSE 0: GO SUB 700
1110 LET l=USR 50000: POKE 50001,40: POKE 50002,160: POKE 50007,0: POKE 50008,64: RETURN
1500 FOR y=3 TO 9: FOR v=3 TO 9
1520 IF a(v,y)=0 OR a(v,y)=2 THEN GO TO 1570
1530 IF a(v,y+1)=1 AND a(v,y+2)=2 THEN RETURN
1540 IF a(v,y-1)=1 AND a(v,y-2)=2 THEN RETURN
1550 IF a(v+1,y)=1 AND a(v+2,y)=2 THEN RETURN
1560 IF a(v-1,y)=1 AND a(v-2,y)=2 THEN RETURN
1570 NEXT v: NEXT y
1575 IF pu=1 AND a(6,6)=1 THEN LET pu=0: FOR o=7 TO 1 STEP -1: BORDER o: GO SUB 700: GO SUB 710: NEXT o: PRINT AT 3,19;pu;" Punkte"
1580 LET h(le)=pu: LET le=le+1
1590 PRINT AT 3,0; FLASH 1;"ENDE": FOR z=1 TO 20: GO SUB 710: NEXT z: LET en=1: RETURN
2000 CLS : PRINT "Leistungskurve","----------------": LET p$="PUNKTE": LET pl=3
2010 FOR z=0 TO 28 STEP 2
2015 IF z>=10 THEN LET pl=2
2020 IF z<=10 THEN PRINT AT z/2+3,0;p$(z/2+1 TO z/2+1)
2030 PRINT AT z/2+3,pl;z: PLOT 35,148-4*z: DRAW 4,0: NEXT z
2045 PLOT 37,148: DRAW 0,-112: LET p$="1 3 5 7 9 1113151719212325": FOR z=1 TO 25 STEP 2: PLOT 38+8*z,27: DRAW 0,4: PRINT AT 19,z+4;p$(z TO z);AT 20,z+4;p$(z+1 TO z+1): NEXT z
2090 PLOT 44,29: DRAW 192,0: PRINT AT 21,25;"SPIEL"
2097 FOR z=1 TO 12: PRINT PAPER 2;AT z+5,5;"                      ": NEXT z
2100 PLOT 44,148-4*h(1): FOR z=1 TO le-2
2105 IF z=25 THEN GO TO 2160
2110 DRAW 8,(h(z)-h(z+1))*4: NEXT z: PRINT AT 21,0; PAPER 6; INK 1; FLASH 1;"ENTER";AT 0,31;""
2160 PAUSE 0: GO SUB 700: RETURN
2500 LET su=1: FOR z=3 TO 9: FOR x=3 TO 9
2502 IF a(x,z)=0 OR a(x,z)=2 THEN GO TO 2515
2504 IF a(x,z+1)=1 AND a(x,z+2)=2 THEN LET zz1=x: LET zz2=z+2: GO SUB 3000
```




```
2505 IF a(x,z-1)=1 AND a(x,z-2)=
2 THEN LET zz1=x: LET zz2=z-2: G
O SUB 3000
2506 IF a(x-1,z)=1 AND a(x-2,z)=
2 THEN LET zz1=x-2: LET zz2=z: G
O SUB 3000
2507 IF a(x+1,z)=1 AND a(x+2,z)=
2 THEN LET zz1=x+2: LET zz2=z: G
O SUB 3000
2515 NEXT x: NEXT z
2520 LET sv=INT (RND*(su-1))+1:
LET b$(1,1)=CHR$ (i(sv,1)+96): L
ET b$(1,2)=CHR$ (i(sv,2)+48): RE
TURN
2600 LET b$(1,1)=CHR$ (i(sv,4)+9
6): LET b$(1,2)=CHR$ (i(sv,3)+48
): RETURN
3000 LET i(su,1)=z-2: LET i(su,2
)=x-2: LET i(su,3)=zz1-2: LET i(
su,4)=zz2-2: LET su=su+1: RETURN


4000 BORDER 1: PAPER 1: INK 6: C
LS : PRINT AT 3,1;"
TAIRE SOLI
4040 PRINT AT 11,8;"Mario Heckma
nn";AT 12,5;"Heilbronner Str. 12
";AT 13,1;"7022 Leinfelden-Echte
rdingen-2";AT 15,10;"© 1 9 8 4"
4080 PRINT AT 21,9; PAPER 6; INK
 1; FLASH 1;"BITTE WARTEN"
4090 SAVE "Solitaire"SCREEN$ : G
O TO 9999
4100 PRINT AT 21,9; INK 1; PAPER
 6; FLASH 1;"    ENTER     ";AT 19
,0;"": PAUSE 0
4120 INPUT "Benoetigen Sie eine
Erklaerung ?(J/N):";t$
4130 IF t$="j" THEN GO TO 5000
4140 GO TO 15
5000 CLS : PRINT "Ihre Aufgabe i
n diesem Spiel istes, durch Spri
ngen mit den blau-en Steinen ein
 verzwicktes Pro-blem zu loesen
.": PRINT
5020 PRINT "Es darf jeweils nu
r ein Steinuebersprungen werde
n, und auchnur dann, wenn hi
nter diesemStein ein weisses Lo
ch frei ist.Ausserdem duerfen S
ie nur senk-recht oder waagerech
t ueber denStein springen. Spr
uenge ueberEck oder diagonal si
nd nicht er-"
5030 PRINT "laubt. Die uebersspru
ngenen Stei-ne werden entfernt.
 Koennen Sienicht mehr springen,
 dann werdendie restlichen Stei
ne als Feh-lerpunkte angerechne
t."
5035 PRINT : PRINT PAPER 6; INK

1; FLASH 1;AT 21,10;"ENTER": PAU
SE 0: CLS : PRINT "Und jetzt zum
 Problem:": PRINT
5040 PRINT "Sie muessen so gesc
hickt spri-ngen, dass erstens n
ur ein Steinuebrigbleibt (1 Punk
t) und zwei-tens dieser Stein am
 Schluss ge-nau im mittleren Loc
h des Spiel-feldes sitzt (0 Punk
te)."
5050 PRINT : PRINT "Damit Sie ei
nen besseren Ueber-blick ueber
den Spielablauf be-kommen, wir
d Sie der Computerdurch das er
ste Spiel fuehren."
5070 PRINT : PRINT "Und nun viel
 Spass!";AT 21,10; PAPER 6; INK
1; FLASH 1;"ENTER": PAUSE 0: LET
 au=1: GO TO 20
6000 RESTORE 9300: FOR g=0 TO 7
6010 READ we: POKE 65504+g,we: N
EXT g: RETURN

6020 RESTORE 9310: FOR g=0 TO 7
6030 READ we: POKE 65504+g,we: N
EXT g: RETURN
6100 IF b$(1,1)="c" AND b$(1,2)=
"o" THEN GO SUB 700: COPY
6110 RETURN
8800 RESTORE : FOR z=1 TO 7: REA
D z$: FOR x=0 TO 7: READ x1: POK
E USR z$+x,x1: NEXT x: NEXT z
8840 FOR z=40000 TO 40185: READ
z1: POKE z,z1: NEXT z: LET l=USR
 40138
8852 RESTORE 9100: FOR z=50000 T
O 50028: READ z1: POKE z,z1: NEX
T z: RETURN
8860 RESTORE 9050: FOR z=3 TO 9:
 FOR x=3 TO 9: READ x1: LET a(z,
x)=x1: NEXT x: NEXT z: RETURN
9000 DATA "a",255,255,127,127,63
,31,15,3,"b",255,255,254,254,252
,248,240,192,"c",3,15,31,63,127,
127,255,255,"d",192,240,248,252,
254,254,255,255
9005 DATA "s",192,48,8,4,2,2,1,1
,"t",1,1,2,2,4,8,48,192,"u",128,
128,64,64,32,16,12,3
9010 DATA 33,193,64,17,8,62,205,
173,156,33,225,73,17,8,62,205,17
3,156,33,196,64,205,173,156,33,2
28,73,17,16,62,205,173,156,33,7,
64,205,173,156,33,167,80,17,24
9020 DATA 62,205,173,156,33,10,6
4,205,173,156,33,170,80,17,32,62
,205,173,156,33,13,64,205,173,15
6,33,173,80,17,40,62,205,173,156
,33,208,64,205,173,156,33,240,73
,17,48,62,205,173,156,33,211,64
9030 DATA 205,173,156,33,243,73,
17,56,62,205,173,156,201,6,8,26,
197,6,4,203,135,31,16,251,119,26
,6,4,203,191,23,16,251,193,35,11
9,43,19,38,16,230,201,33,120,255
,17,136,61,14,0,205,232,156,205,
241,156,205,241,156,205,232,156,

62,1,129,79,62,7,185,32
9040 DATA 235,201,6,4,62,0,119,3
5,16,250,201,6,4,26,119,35,19,16
,250,201
9050 DATA 0,0,1,1,1,0,0,0,0,1,1,
1,0,0,1,1,1,1,1,1,1,1,1,2,1,1,
1,1,1,1,1,1,1,1,0,0,1,1,1,0,0,0,
0,1,1,1,0,0
9100 DATA 17,40,160,1,0,27,33,0,
64,126,18,35,19,11,62,0,184,40,2
,24,244,62,0,185,40,2,24,237,201
9300 DATA 3,12,16,32,64,64,128,1
28
9310 DATA 8,16,16,0,0,0,0,0
9999 SAVE "Solitaire": STOP
```





PAINTER ist ein Malprogramm für die Grafikstufe 8. Es benötigt mindestens 32 KB RAM.

Die Bilder des Painter sind kompatibel zu anderen Malprogrammen wie Micro-Painter oder Grafik-Zauberer, jedoch nicht zum Koalapad (Micro Illustrator).

Nach dem Programmstart sieht man kurz, wie das Menue ausgedruckt wird, dann wird zur Grafik umgeschaltet und nach etwa 3 Sekunden erscheint der Cursor in der Mitte des Bildschirms. Er läßt sich mit dem Joystick (Buchse 1) bewegen. Wenn man eine Weile mit dem Cursor fährt, dann wird er schneller.

Drücken Sie die Taste „1", dann können Sie durch Druck auf den Feuerknopf des Joysticks Punkte setzen und Linien ziehen. Nachdem Sie die Taste „0" gedrückt haben können Sie wieder löschen.

Mit der Taste „RETURN" können Sie zwischen Grafik und Menue-Anzeige umschalten.

Folgende Funktionen stehen zur Verfügung:

0,1: Farbe, mit der der Cursor malt.
2: Linie ziehen. Von dem Punkt, an dem zuletzt der Feuerknopf gedrückt wurde bis zum Cursor wird eine Linie gezogen.
3: Parallele ziehen. Es wird eine Parallele zur letzten Linie gezogen. Der Cursor ist dabei der eine Punkt, der andere wird errechnet.
4: Rechts füllen. Der Computer führt ein DRAWTO aus (wie 2) und verlängert jeden Punkt der Linie nach rechts bis er auf einen Punkt der Farbe 1 stößt.
5: Kreis zeichnen. Der Punkt, an dem zuletzt der Feuerknopf gedrückt wurde ist der Mittelpunkt, der Cursor liegt auf der Kreislinie.
6: Bildschirm löschen.
7: Farben setzen. Nach drücken von „7" geben Sie ein, welches Farbregister Sie ändern wollen (0-4). Dann können Sie mit dem Joystick die Farbe (links-rechts) und die Helligkeit (vorn-hinten) bestimmen. Wenn Sie fertig sind, drücken Sie den Feuerknopf und wählen ein neues Register oder gehen Sie mit der „Space"-Taste in den Normalmodus.
8: Info-Zeile ein/aus. Über dem Grafikbildschirm werden die Position des Cursors (X und Y) sowie die eingeschaltete Farbe angezeigt.
9: Sprung zum Disk-Menue.

Im Disk-Menue wird der, der eine Diskettenstation hat ohne weiteres zurechtkommen.

Für Besitzer von Cassettenrecordern sind nur die Funktionen 2 und 3 interessant (Bild laden, Bild abspeichern).

Wenn Sie ein Bild auf Cassette speichern oder von dort laden wollen, dann müssen Sie nach drücken von „2" bzw. „3" das angebotene „D1:" mit der Taste „DELETE/BACKS" löschen und „C:" und irgendeinen Namen (z. B. „BILD" schreiben. Das ist wichtig, sonst kommt ERROR 5 (String zu kurz). Der Name kann beim Speichern auf Cassette bei allen Bildern gleich sein.

```
10 REM **********************************************************************
20 REM *                                                                    *
30 REM *                              PAINTER                               *
40 REM *                              ^^^^^^^^^^^                           *
50 REM *                      von ROLAND REYER                              *
60 REM *                          AM GIEGENBERG 21, 6438 LUDWIGSAU 1        *
70 REM !                              5/84                                  *
80 REM *                                                                    *
90 REM **********************************************************************
110 REM [         INITIALISIERUNG        ]
120 POKE 106,PEEK(106)-36
130 GRAPHICS 0:OPEN #3,9,0,"E:"
140 D0L=PEEK(560):D0H=PEEK(561):GR=8
150 M0L=PEEK(88):M0H=PEEK(89)
160 GOSUB 1260
170 POKE 106,PEEK(106)+36
180 OPEN #1,4,0,"K:":GRAPHICS 24:DIM A$(40),FARB(4),HELL(4),CIO$(28)
190 M8L=PEEK(88):M8H=PEEK(89)
200 GOSUB 1200
210 FOR J=0 TO 4:FARB(J)=INT(PEEK(708+J)/16):HELL(J)=PEEK(708+J)-FARB(J)*16:NEXT
 J
220 REM [ ASSEMBLER-ROUTINE ZUM AUF- ]
230 REM [ RUFEN VON *CIO*            ]
240 RESTORE 260
250 FOR J=1 TO 28:READ X:CIO$(J)=CHR$(X):NEXT J
260 DATA 104,104,104,170,104,104
270 DATA 157,66,3,104,157,69,3
280 DATA 104,157,68,3,104,157,73
290 DATA 3,104,157,72,3,76,86,228
300 X=160:Y=96:X1=160:Y1=96
310 REM [    JOYSTICK+GRAPHIK-MENUE     ]
320 K=STICK(0):Q=STRIG(0):IF Q=1 AND K<15 THEN ST=ST+0.2-0.2*(ST>4):GOTO 340
330 ST=1
340 X=X+INT(ST*(((K=5 OR K=6 OR K=7) AND (X<319))-((K=9 OR K=10 OR K=11) AND (X>
0)))+0.2)
350 Y=Y+INT(ST*(((K=5 OR K=9 OR K=13) AND (Y<191))-((K=6 OR K=10 OR K=14) AND (Y
>0)))+0.2)
360 IF Q=0 THEN Z=COL
```

```
370 IF X=X1 AND Y=Y1 THEN 390
380 TRAP 1140:COLOR Z:PLOT X1,Y1:LOCATE X,Y,Z:X1=X:Y1=Y
390 TRAP 400:IF PEEK(764)<>255 THEN GET #1,M:M=M-47:ON M GOSUB 450,460,480,500,5
20,560,660,680,790,1420
400 COLOR ABS(Z-1):PLOT X,Y:IF Q=0 THEN XM=X:YM=Y
410 IF INF=1 THEN 1060
420 IF M=108 THEN GOSUB 1020
430 GOTO 310
440 REM [        FABEN EINSCHALTEN       ]
450 COL=0:RETURN
460 COL=1:RETURN
470 REM [           LINIE ZIEHEN         ]
480 COLOR COL:PLOT X,Y:DRAWTO XM,YM:XZ=X:YZ=Y:Z=COL:RETURN
490 REM [        PARRALLELE ZIEHEN       ]
500 TRAP 590:XM=XM+(X-XZ):YM=YM+(Y-YZ):GOTO 480
510 REM [           FILL-BEFEHL          ]
520 POSITION XM,YM:POKE 765,COL
530 XIO 18,#6,0,0,"S:"
540 GOSUB 1240:RETURN
550 REM [          KREIS ZEICHNEN        ]
560 COLOR COL:R=SQR((X-XM)^2+(Y-YM)^2):XK1=XM:YK1=YM+R:FOR J=0 TO 6.28 STEP 0.02
570 TRAP 580:XK=SIN(J)*R+XM:YK=COS(J)*R+YM:PLOT XK1,YK1:DRAWTO XK,YK
580 XK1=XK:YK1=YK:NEXT J:Z=COL:RETURN
590 XMQ=XM:YMQ=YM
600 IF XMQ<0 THEN YMQ=Y+((YM-Y)*(X/(X-XM))):XMQ=0:GOTO 630
610 IF YMQ<0 THEN XMQ=X+((XM-X)*(Y/(Y-YM))):YMQ=0:GOTO 630
620 GOTO 640
630 TRAP 600:DRAWTO XMQ,YMQ
640 Z=COL:XZ=X:YZ=Y:RETURN
650 REM [       BILDSCHIRM LOESCHEN      ]
660 ? #6,"*F*";:Z=0:RETURN
670 REM [        FARBEN VERAENDERN       ]
680 GET #1,F:F=F-48
690 IF F=-16 THEN RETURN
700 IF F<0 OR F>4 THEN 680
710 K=STICK(0)
720 FARB(F)=FARB(F)+((K=5 OR K=6 OR K=7) AND (FARB(F)<16))-((K=9 OR K=10 OR K=11
) AND (FARB(F)>0))
730 HELL(F)=HELL(F)+((K=5 OR K=9 OR K=13) AND (HELL(F)<15))-((K=6 OR K=10 OR K=1
4) AND (HELL(F)>0))
740 SETCOLOR F,FARB(F),HELL(F)
750 FOR T=1 TO 20:NEXT T
760 IF STRIG(0)=0 THEN 680
770 GOTO 710
780 REM [       INFO-ZEILE EIN/AUS       ]
790 POKE 559,0:IF INF=1 THEN GOTO 910
800 A=D8L+D8H*256
810 POKE A,66
820 POKE A+1,M8L+16
830 POKE A+2,M8H+30
840 POKE A-1,112
850 POKE A-2,112
860 A=A-2
870 D8H=INT(A/256):D8L=A-D8H*256
880 INF=1:POKE 703,4:? "*F*":POKE 703,24
890 GOTO 980
900 REM [         INFO-ZEILE AUS         ]
910 A=D8L+D8H*256
920 A=A+2
930 D8H=INT(A/256):D8L=A-D8H*256
940 POKE A,112
950 POKE A+1,112
```



```
960 POKE A+2,112
970 INF=0
980 GOSUB 1040:GR=8
990 POKE 559,34
1000 RETURN
1010 REM [     UMSCHALTEN ZWISCHEN     ]          [       GRAPHIK UND MENUE      ]
1020 M=10:IF GR=0 THEN GR=8:GOTO 1040
1030 POKE 560,D0L:POKE 561,D0H:GR=0:RETURN
1040 POKE 560,D8L:POKE 561,D8H
1050 RETURN
1060 REM [   INFO-ZEILE SCHREIBEN      ]
1070 POKE 752,1:POKE 703,4
1080 ? "¥B¥";
1090 ? "X=";X;"  ¥H¥";
1100 ? "Y=";Y;"  ¥H¥";
1110 ? "COL=";COL
1120 POKE 752,0:POKE 703,24
1130 POSITION X,Y:GOTO 420
1140 REM [    KOORDINATEN-KORREKTUR    ]
1150 IF X<0 THEN X=0
1160 IF X>319 THEN X=319
1170 IF Y<0 THEN Y=0
1180 IF Y>191 THEN Y=191
1190 GOTO 380
1200 REM [    DLIST-KORREKTUR          ]
1210 D8L=PEEK(560):D8H=PEEK(561)
1220 RETURN
1230 REM [     FARBWERTE EINSETZEN     ]
1240 GRAPHICS 56:FOR J=0 TO 4:SETCOLOR J,FARB(J),HELL(J):NEXT J:IF INF=1 THEN GO
SUB 1200:POKE 559,0:GOSUB 800
1250 GR=8:RETURN
1260 REM [        MENUE DRUCKEN        ]
1270 ? "¥F¥":? "¥HH¥[ MENUE ]":?
1280 ? "¥H¥[0]=LOESCHEN"
1290 ? "¥H¥[1]=SCHREIBEN"
1300 ? "¥H¥[2]=LINIE ZIEHEN"
1310 ? "¥H¥[3]=PARALLELE ZIEHEN"
1320 ? "¥H¥[4]=RECHTS FUELLEN"
1330 ? "¥H¥[5]=KREIS ZEICHNEN"
1340 ? "¥H¥[6]=BILDSCHIRM LOESCHEN"
1350 ? "¥H¥[7]=FARBEN SETZEN"
1360 ? "¥H¥[8]=INFO-ZEILE"
1370 ? "¥H¥[9]=[DISK-MENUE]"
1380 ? :? "[ RETURN ]=GRAPHIK <-> MENUE - WECHSEL":? "[ ¥D¥]";
1390 GOSUB 2480
1400 RETURN
1410 REM [        DISK MENUE           ]
1420 COLOR Z:PLOT X1,Y1
1430 GOSUB 2460:POKE 87,0
1440 GR=8:GOSUB 1020
1450 POKE 106,PEEK(106)-36
1460 GRAPHICS 0:SETCOLOR 1,0,14:SETCOLOR 2,3,5:SETCOLOR 4,0,0
1470 ? "¥FCHHD¥[ DISK-MENUE ]"
1480 ? :? "¥H¥[1]=DIREKTORY"
1490 ? :? "¥H¥[2]=BILD LADEN"
1500 ? :? "¥H¥[3]=BILD ABSPEICHERN"
1510 ? :? "¥H¥[4]=DATEI SICHERN"
1520 ? :? "¥H¥[5]=DATEI ENTSICHERN"
1530 ? :? "¥H¥[6]=DATEI UMBENENNEN"
1540 ? :? "¥H¥[7]=DATEI LOESCHEN"
1550 ? :? "¥H¥[8]=[ GRAPHIK-MENUE ]"
1560 ?
```




```
1570 GET #1,MEN
1580 REM
1590 MEN=MEN-48
1600 TRAP 1570
1610 ON MEN GOTO 1630,1690,1830,1950,2020,2090,2160,2360
1620 GOTO 1570
1630 REM [         DIREKTORY              ]
1640 TRAP 1670:POSITION 2,20:? "D1:*.*¥DDD¥";:GOSUB 2290:? "¥FC¥";A$
1650 OPEN #2,6,0,A$
1660 INPUT #2,A$:? A$:GOTO 1660
1670 CLOSE #2:? :? "[  DRUECKEN SIE EINE TASTE  ]":GET #1,A:GOTO 1470
1680 REM [         BILD LADEN             ]
1690 POSITION 5,5:? "¥AE¥";
1700 GOSUB 2280
1710 TRAP 2410
1720 IF A$="" OR A$(4)="" THEN 1810
1730 SRAM=M8L+M8H*256
1740 OPEN #2,4,0,A$
1750 Q=USR(ADR(CIO$),32,7,SRAM,7680)
1760 FOR J=0 TO 4
1770 GET #2,FARB:GET #2,HELL
1780 FARB(J)=FARB:HELL(J)=HELL
1790 NEXT J
1800 CLOSE #2:IF PEEK(867)<>1 THEN POKE 195,PEEK(867):GOTO 2410
1810 GOTO 1470
1820 REM [       BILD ABSPEICHERN         ]
1830 POSITION 5,7:? "¥AE¥";
1840 GOSUB 2280
1850 TRAP 2410
1860 IF A$="" OR A$(4)="" THEN 1810
1870 SRAM=M8L+M8H*256
1880 OPEN #2,8,0,A$
1890 Q=USR(ADR(CIO$),32,11,SRAM,7680)
1900 FOR J=0 TO 4
1910 PUT #2,FARB(J):PUT #2,HELL(J)
1920 NEXT J
1930 CLOSE #2:IF PEEK(867)<>1 THEN POKE 195,PEEK(867):GOTO 2410
1940 GOTO 1470
1950 REM [        DATEI SICHERN           ]
1960 POSITION 5,9:? "¥AE¥";
1970 GOSUB 2280
1980 TRAP 2410
1990 IF A$="" OR A$(4)="" THEN 1470
2000 XIO 35,#2,0,0,A$
2010 GOTO 1470
2020 REM [        DATEI ENTSICHERN        ]
2030 POSITION 5,11:? "¥AE¥";
2040 GOSUB 2280
2050 TRAP 2410
2060 IF A$="" OR A$(4)="" THEN 1470
2070 XIO 36,#2,0,0,A$
2080 GOTO 1470
2090 REM [        DATEI UMBENENNEN        ]
2100 POSITION 5,13:? "¥AE¥";
2110 GOSUB 2280
2120 TRAP 2410
2130 IF A$="" OR A$(4)="" THEN 1470
2140 XIO 32,#2,0,0,A$
2150 GOTO 1470
2160 REM [       DATEI LOESCHEN           ]
2170 POSITION 5,15:? "¥AE¥";
2180 GOSUB 2280
```

```
2190 TRAP 2410
2200 IF A$="" OR A$(4)="" THEN 1470
2210 OPEN #2,6,0,A$:CLOSE #2:? "%BIB%";
2220 ? "DRUECKEN SIE -Y- ZUM LOESCHEN VON.."? "  --->";A$:?
2230 ? " [ACHTUNG:ALLE DATEIEN MIT DIESER]       [SPEZIFIKATION WERDEN GELOESCHT
]!%D%";
2240 GET #1,A:IF A<>89 THEN 1470
2250 XIO 33,#2,0,0,A$
2260 GOTO 1470
2270 REM [       EINGABE ROUTINE       ]
2280 POSITION 2,20:? "D1:";
2290 GET #1,A
2300 IF A=156 OR A=157 OR A=125 OR A=28 OR A=29 THEN 2290
2310 ? CHR$(A);
2320 IF A<>155 THEN 2290
2330 POSITION 2,20
2340 INPUT #3,A$
2350 RETURN
2360 REM [  SPRUNG ZUM GRAPHIK-MENUE  ]
2370 GOSUB 1270:POKE 106,160
2380 GOSUB 1240
2390 POP :LOCATE 160,96,Z:GOTO 300
2400 REM [       FEHLERMELDUNG        ]
2410 CLOSE #2:POSITION 2,21:? "%III%";
2420 ? "ERROR - ";PEEK(195)
2430 ? :? "[ DRUECKEN SIE EINE TASTE]";
2440 GET #1,A:GOTO 1470
2450 REM [     ZEIGER F. SCREEN-RAM     ]
2460 POKE 88,M0L:POKE 89,M0H:POKE 87,0
2470 RETURN
2480 POKE 88,M8L:POKE 89,M8H:POKE 87,8
2490 RETURN
2500 REM
2510 REM
2520 REM ERKLAERUNG ZUR EINGABE DER SONDERZEICHEN
2530 REM
2540 REM
2550 REM -ALLES, WAS IN ECKIGEN KLAMMERN ([]) STEHT, WIRD INVERS GESCHRIEBEN
2560 REM
2570 REM - ALLES, WAS ZWISCHEN %-ZEICHEN STEHT, SIND SONDERZEICHEN !!
2580 REM DIESE ZEICHEN WERDEN WIE FOLGT EINGEGEBEN:
2590 REM
2600 REM -A=ESC,ESC (2*DRUECKEN)
2610 REM -B=ESC,CTRL-MINUS (PFEIL NACH OBEN)
2620 REM -C=ESC,CTRL-GLEICH (PFEIL NACH UNTEN)
2630 REM -D=ESC,CTRL-PLUS (PFEIL NACH LINKS)
2640 REM -E=ESC,CTRL-MAL (PFEIL NACH RECHTS)
2650 REM -F=ESC,SHIFT-CLEAR (SCHRAEGER PFEIL NACH OBEN)
2660 REM -G=ESC,DELETE (DREIECK NACH LINKS)
2670 REM -H=ESC,TAB (DREIECK NACH RECHTS)
2680 REM -I=ESC,SHIFT-DELETE (WIE "B", JEDOCH INVERS)
2690 REM -J=ESC,SHIFT-INSERT (WIE "C", JEDOCH INVERS)
2700 REM -K=ESC,SHIFT-TAB (WIE "D", JEDOCH INVERS)
2710 REM -L=ESC,CTRL-TAB (WIE "E", JEDOCH INVERS)
2720 REM -M=ESC,CTRL-2 (WIE "F", JEDOCH INVERS)
2730 REM -N=ESC,CTRL-DELETE (WIE "G", JEDOCH INVERS)
2740 REM -O=ESC,CTRL-INSERT (WIE "H", JEDOCH INVERS)
2750 REM
2760 REM
2770 REM -BEISPIEL: %CD%  = ESC,CTRL-=,ESC,CTRL-+
```



Hardcopy für Graphics 7+ und 8

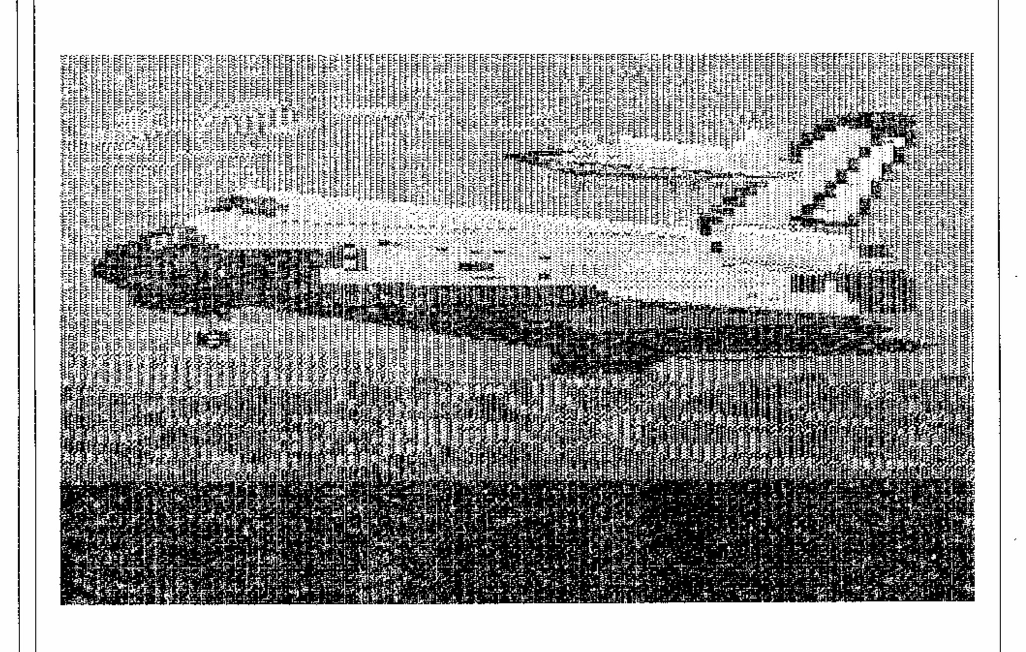
Dieses Programm ermöglicht es, Bilder des Graphics Mode 7+ (7E) und 8 auf dem Drucker auszugeben. Die Bilddaten müssen über die Handler (IOCB) ladbar sein. Daher können als Datenträger sowohl Diskette als auch Kassette benutzt werden. Dieses Programm macht das Hardcopy in einer Minute. Die Laderoutine des Programms ist MICRO-PAINTER (tm) kompatibel.

SYSTEM REQUIREMENT:
ATARI 400/800/600XL/800XL Computer
ATARI 810/1050 Disk Drive
ATARI 410/1010 Program Recorder
ATARI (EPSON RX../MX../RX..)

Laden des Programmes:

I. Für Kassete
1. Programmkassette einlegen
2. Mit CLOAD laden
3. Mit RUN starten
4. Daten Kassette oder Diskette einlegen

II. Für Diskette
1. Programmdiskette einlegen
2. Mit LOAD "D:GR8PRINT.BAS" laden
3. Mit RUN starten
4. Daten Diskette oder Kassette einlegen



Als erstes erscheint das Einführungsbild, das mit START verlassen werden kann. Dann fragt der Computer, ob er die Directory der Disk ausgeben soll. Dies ist aber nur möglich, wenn eine Disk angeschlossen ist. Ansonsten tritt eine Fehlermeldung auf, die dann nach Betätigen der START- Taste zu einem erneuten Start des Programmes führt. Die Frage nach der Directory muß mit '1' oder '0' beantwortet werden. '1' bezweckt die Ausgabe. Anschließend muß der Bildname eingegeben werden (Bsp. D:TIGER.MIC). Nun erfolgt das Laden des Bildes. Nach dem Ladevorgang führt ein Betätigen der START- Taste zum Ausdruck des Bildes. Ein erneutes Drücken zum Neustart.

```
0 REM ************************
1 REM *                      *
2 REM * HARDCOPY fuer 7+ und 8 *
3 REM *                      *
4 REM *   by Klaus Rupprecht  *
5 REM *                      *
6 REM *  (c) 1984  20/05/1984 *
7 REM *                      *
8 REM ************************
9 REM
10 GRAPHICS 17:? #6:? #6;"[illegible]"
11 ? #6:? #6:? #6;"    hardcopy fuer"
12 ? #6:? #6:? #6:? #6;"  graphics 7+ und 8"
13 ? #6:? #6:? #6:? #6;"         BY"
14 ? #6:? #6:? #6:? #6;"  [illegible]"
20 ? #6:? #6:? #6:? #6;"[illegible]"
90 IF PEEK(53279)<>6 THEN 90
99 CLR :DIM B$(16),A$(192),X$(17):TRAP 1000
```



```
100 GRAPHICS 0:? "↰↓↓→→→DIRECTORY OF DISK (0/1)";:INPUT Y:IF Y=1 THEN 1200
101 ? "↰↓↓↓↓↓→→→→Geben Sie den Bildernamen ein.":? "↓↓→→→→→→→→";:INPUT B$
102 IF B$(1,1)<>"D" AND B$(1,1)<>"C" THEN 101
110 ? "↓↓↓→→→→GR 8 ODER 7+ ":? "↓↓→→→→Bei GR 8 0 eingeben.":? "↓→→→→→→→";
:INPUT Q
130 GRAPHICS 8+16
150 DPL=PEEK(560)+256*PEEK(561)
160 DPM=PEEK(DPL+4)+256*PEEK(DPL+5)
171 IF Q=0 THEN 300
200 POKE DPL+3,78
210 FOR A=0 TO 92:POKE DPL+6+A,14:NEXT A:POKE DPL+6+93,78
220 FOR A=96 TO 192:POKE DPL+6+A,14:NEXT A
260 TRAP 500
300 OPEN #1,4,0,B$
340 HAND=832+16
350 POKE HAND+2,7
390 POKE HAND+5,INT(DPM/256):POKE HAND+4,INT(DPM-(256*PEEK(HAND+5)))
410 BYTES=7680:POKE HAND+9,INT(BYTES/256):POKE HAND+8,INT(BYTES-(256*PEEK(HAND+9
)))
440 TRAP 1500
450 DUMMY=USR(ADR("hhh▒LV▒"),16)
460 TRAP 1000
481 IF Q=0 THEN 500
490 GET #1,A:POKE 712,A:FOR A=0 TO 2:GET #1,B:POKE 708+A,B:NEXT A
500 CLOSE #1
510 IF PEEK(53279)<>6 THEN 510
600 RESTORE 660:FOR B=1 TO 61:READ N:POKE 1535+B,N:NEXT B:SM=PEEK(88)+256*PEEK(8
9):SM=SM+40*191
610 LPRINT CHR$(27);"A";CHR$(8):FOR X=SM TO SM+39
620 A$=CHR$(0):A$(192)=CHR$(0):A$(2)=A$
640 W=USR(1536,X,ADR(A$))
641 LPRINT "␛K▒▒";A$
650 NEXT X
655 IF PEEK(53279)<>6 THEN 655
656 RUN
660 DATA 104,104,141,21,6,104,141,20,6,104,141,27,6,104,141,26,6,160,193,173,255
,255,136,240,35,141,255,255,238
670 DATA 26,6,240,21,173,20,6,56,233,40,141,20,6,144,4,24,76,19,6,206,21,6,76,19
,6,238,27,6,76,33,6,96
1000 GRAPHICS 0:IF PEEK(195)=138 THEN ? "↓↓↓↓↓↓PRINTER oder DISK antwortet n
icht!!"
1001 ? "↓↓Druecken Sie die START-Taste.":GOTO 1010
1002 ? "↰↓↓↓→→→→ERROR :";PEEK(195)
1010 IF PEEK(53279)<>6 THEN 1010
1020 RUN
1200 CLOSE #1:OPEN #4,6,0,"D:*.*"
1210 INPUT #4,X$:? X$
1220 IF X$(5,8)="FREE" THEN 1240
1230 GOTO 1210
1240 IF PEEK(53279)<>6 THEN 1240
1250 GOTO 101
1500 IF PEEK(195)=136 THEN 460
1510 GOTO 1000
```



Zyklo

Ein Labyrinthspiel für die Grundversion des VC-20. Ziel ist es, in die Mitte des Labyrinths bzw. des Bildschirmes zu gelangen. Achten Sie auf dem Weg dorthin auf die sich zufällig öffnenden Türen. Sie müssen so schnell wie möglich durchlaufen werden. Schließt sich eine Tür und Sie befinden sich gerade im Durchgang, dann ist das Spiel zu Ende.

Gefahr droht Ihnen während des Spiels jedoch noch von anderer Seite. Im Labyrinth tummeln sich Monster und Gespenster. Alle machen Jagd auf Sie und versuchen mit allen Mitteln Ihnen den Weg abzuschneiden. Vorsicht! – Ein Gespenst ist dabei besonders hartnäckig.



Versuchen Sie also, so weit wie möglich ins Labyrinth vorzudringen und den Verfolgern geschickt auszuweichen. Eine kleine Hilfe: Im Spiel verteilt sind insgesamt vier Früchte von denen Sie essen können. Nach dem Verzehr sind Sie kurzfristig in der Lage eine Mauer zu sprengen, wenn dies erforderlich ist.

Steuerung: links >L<, aufwärts >p<, rechts >;<, abwärts >Leertaste<. Mit >:< können Sie die Mauer sprengen.

```
1 PRINT"▮▮":POKE36879,10:POKE36878,15:E=7680
10 FORT=38TO150:POKE36865,T:FORTT=0TO10:NEXT:NEXT
11 PRINT"▮▮▮▮▮▮▮▮▮SPIELANLEITUNG?(J/N)"
12 FORT=150TO38STEP-1:POKE36865,T:FORTT=0TO10:NEXT:NEXT
14 GETA$:IFA$=""GOTO14
15 IFA$="N"GOTO113
16 FORT=0TO1:FORTT=140TO240:POKE36876,TT:NEXT:NEXT:POKE36876
,0
20 PRINT"▮"
22 PRINT"DAS ZIEL IST ES, ZUM":PRINT
24 PRINT"SCHATZ IN DER MITTE ":PRINT
26 PRINT"DES BILDSCHIRMS ZU GE-"
28 PRINT"LANGEN. DOCH MONSTER,":PRINT
30 PRINT"GEISTER UND TUEREN ":PRINT
32 PRINT"HINDERN SIE DARAN.":PRINT
34 FORT=0TO21:POKEE+15*22+T,83:POKE38400+15*22+T,3:POKE36877
,200:FORTT=0TO29:NEXT
36 POKE36877,0:NEXT
38 PRINT"▮▮▮▮▮▮▮▮▮▮▮▮▮-TASTE-":POKE198,0:WAIT198,1
40 PRINT"▮"
42 PRINT"DIE TUEREN OEFFNEN UND"
44 PRINT"SCHLIESSEN SICH VON":PRINT
46 PRINT"ZEIT ZU ZEIT. ":PRINT
48 PRINT"▮▮VORSICHT! WENN SICH ":PRINT
50 PRINT"DIE TUER SCHLIESST UND"
52 PRINT"SIE BEFINDEN SICH IM ":PRINT
54 PRINT"DURCHGANG, IST DAS ":PRINT
56 PRINT"SPIEL ZU ENDE.":PRINT
58 PRINT"▮▮▮▮▮▮▮▮▮▮-TASTE-":POKE198,0
60 FORT=0TO21:POKEE+7*22+T,81:POKE38400+7*22+T,4:POKE36877,2
00:POKE36877,0:NEXT:WAIT198,1
62 PRINT"▮"
64 FORT=150TO128STEP-4:POKE36866,T:FORTT=0TO50:NEXT:NEXT
66 PRINT"WENN SIE EINE DER ":PRINT
68 PRINT"FRUECHTE AM RAND FRES-"
70 PRINT"SEN, SO KOENNEN SIE":PRINT
72 PRINT"EINMAL DIE MAUER SPREN"
74 PRINT"-GEN.":PRINT
76 PRINT"DIES MACHEN SIE INDEM":PRINT
78 PRINT"SIE ´:´ UND DANN DIE":PRINT
80 PRINT"GEWUENSCHTE RICHTUNG":PRINT
82 PRINT"EINTASTEN."
83 FORT=128TO150STEP4:POKE36866,T:FORTT=0TO50:NEXT:NEXT:POKE
36866,150
84 PRINT"▮▮▮▮▮▮▮▮▮-TASTE-":POKE198,0:WAIT198,1
86 POKE36865,150:PRINT"▮"
88 PRINT"▮STEUERUNG:"
```

```
90 PRINT"────────────────":PRINT
92 PRINT"P"
94 PRINT"|"
96 PRINT"L-+-;"
98 PRINT"|  "
99 FORT=150TO38STEP-1:POKE36865,T:FORTT=0TO10:NEXT:NEXT
100 PRINT"SPACE"
101 PRINT"-TASTE-":POKE198,0:WAIT198,1:PRINT""
102 PRINT"WENN SIE NUN EINE":PRINT
104 PRINT"TASTE DRUECKEN,WIRD ":PRINT
106 PRINT"DAS PROGRAMM GELOESCHT"
108 PRINT"UND SIE MUESSEN LOAD":PRINT
110 PRINT"EINGEBEN."
112 POKE198,0:WAIT198,1
113 PRINT"BITTE ' LOAD '"
1000 NEW
1001 REM (C) BY JOHANN
1002 REM STRAUSS JUN.
1003 REM OSTRIACH 102
1004 REM 9570 OSSIACH
```

Teil 2

```
0 POKE36876,200
1 READA:IFA=-1GOTO20
2 POKE7168+T+8*40,A:T=T+1:GOTO1
20 FORT=0TO7:POKE7168+32*8+T,0:NEXT
30 PRINT"":POKE36876,0:PRINT"NOCHMALS ' LOAD
 '":NEW
100 DATA24,126,24,60,126,90,24,60,0,127,127,127,127,127,127,
127,129,66,60,90,126,60,24
110 DATA60,24,60,90,126,126,90,165,0,255,129,129,129,129,129
,129,255,1,2,4,106,127,125
115 DATA60,24,129,126,102,90,90,102,126,129,-1
```

Teil 3

```
1 PRINT"":DIMA$(11):E=7680:Y=66:X=5:POKE36878,15:DIMO(23):
POKE36869,255:Q(0)=210:POKE36879,14
2 Q(1)=190:Q(2)=122:YY=418::FORT=0TO10:READA$(T):PRINT"□"A$(
T):NEXT:FORT=9TO0STEP-1:PRINT"□"A$(T):NEXT
15 FORT=0TO23:READO(T):POKE38400+O(T),3:NEXT:XX=10:GOSUB300
100 FORT1=0TO2:POKE36876,0
105 PE=PEEK(197):IFPE=13THENGOSUB500:GOSUB300
110 IFPE=32THENGOSUB500:GOSUB310
115 IFPE=21THENGOSUB500:GOSUB320
120 IFPE=22THENGOSUB500:GOSUB330
121 IFR=0GOTO125
122 IFPE=45THENR=1:GOSUB1000
125 A=INT(RND(1)*4)+1:POKEE+Q(T1),32:ONAGOSUB400,410,420,430
:G=G+1:IFG>8THENG=0:GOSUB590
135 W=XX:W1=YY:I=I+1:IFI>1THENI=0:GOSUB800
137 IFXX=WANDYY=W1THENI1=I1+1:GOTO139
138 I1=0:POKEE+231,54:POKE38631,4
139 IFI1>2THENGOSUB1200
140 POKEE+XX+YY,50:POKE38400+XX+YY,7:IFX=11ANDY=220THEND=1:P
=P+100:GOTO2000
245 NEXT:GOTO100
300 Y=Y-22:CI=PEEK(E+X+Y):IFCI=49ORY<0ORCI=52THENY=Y+22
302 IFCI=53THENPOKE36876,240:R=1
303 IFCI=51GOTO2000
305 POKEE+X+Y,48:POKE38400+Y+X,1:RETURN
```



```
310 Y=Y+22:CI=PEEK(E+X+Y):IFCI=49ORCI=52ORY>482THENY=Y-22
312 IFCI=53THENPOKE36876,240:R=1
313 IFCI=51GOTO2000
315 POKEE+X+Y,48:POKE38400+X+Y,1:RETURN
320 X=X-1:CI=PEEK(E+X+Y):IFCI=49ORCI=52ORX<0THENX=X+1
322 IFCI=53THENPOKE36876,240:R=1
323 IFCI=51GOTO2000
325 POKEE+Y+X,48:POKE38400+Y+X,1:RETURN
330 X=X+1:CI=PEEK(E+Y+X):IFCI=49ORCI=52ORX>21THENX=X-1
332 IFCI=53THENPOKE36876,240:R=1
333 IFCI=51GOTO2000
335 POKEE+Y+X,48:POKE38400+Y+X,1:RETURN
400 Q(T1)=Q(T1)-22:IFPEEK(E+Q(T1))=48GOTO2000
405 IFPEEK(E+Q(T1))<>32THENQ(T1)=Q(T1)+22
409 POKEE+Q(T1),51:POKE38400+Q(T1),5:RETURN
410 Q(T1)=Q(T1)+22:IFPEEK(E+Q(T1))=48GOTO2000
415 IFPEEK(E+Q(T1))<>32THENQ(T1)=Q(T1)-22
419 POKEE+Q(T1),51:POKE38400+Q(T1),5:RETURN
420 Q(T1)=Q(T1)-1:IFPEEK(E+Q(T1))=48GOTO2000
425 IFPEEK(E+Q(T1))<>32THENQ(T1)=Q(T1)+1
429 POKEE+Q(T1),51:POKE38400+Q(T1),5:RETURN
430 Q(T1)=Q(T1)+1:IFPEEK(E+Q(T1))=48GOTO2000
435 IFPEEK(E+Q(T1))<>32THENQ(T1)=Q(T1)-1
439 POKEE+Q(T1),51:POKE38400+Q(T1),5:RETURN
500 POKEE+X+Y,32:POKE36876,200:RETURN
590 IFPEEK(E+O(01))=48ORPEEK(E+O(02))=48GOTO2000
600 POKEE+O(01),52:POKEO(02)+E,52:POKE36876,180:O1=INT(RND(1
)*23):O2=INT(RND(1)*24)
605 POKEE+O(01),32:POKEE+O(02),32:P=P-2:RETURN
800 POKEE+XX+YY,32:IFYY<YTHENYY=YY+22:GOSUB900:IFJTHENYY=YY-
22
805 IFYY>YTHENYY=YY-22:GOSUB900:IFJTHENYY=YY+22
810 IFXX>XTHENXX=XX-1:GOSUB900:IFJTHENXX=XX+1
815 IFXX<XTHENXX=XX+1:GOSUB900:IFJTHENXX=XX-1
817 RETURN
900 IFPEEK(E+XX+YY)=48GOTO2000
905 IFPEEK(E+XX+YY)<>32THENJ=1:RETURN
910 J=0:RETURN
1000 POKE36876,230:GOSUB1100:P=P+10:POKEE+Y+X+G,32:R=0:RETUR
N
1100 SS=PEEK(197):IFSS=32THENG=+22:RETURN
1105 IFSS=13THENG=-22:RETURN
1110 IFSS=21THENG=-1:RETURN
1115 IFSS=22THENG=1:RETURN
1120 GOTO1100
1200 U=INT(RND(1)*4)-1:IFU=0THENU=22:GOTO1400
1205 IFU=2THENU=-22:GOTO1400
1300 XX=XX+U:CI=PEEK(E+XX+YY):IFCI=49ORCI=52ORXX>21ORXX<0THE
NXX=XX-U:RETURN
1305 POKEE+XX+YY,50:POKEE+XX-U+YY,32:GOTO1300
1400 YY=YY+U:CI=PEEK(E+XX+YY):IFCI=49ORCI=52ORYY>484ORYY<0TH
ENYY=YY-U:RETURN
1405 POKEE+XX+YY,50:POKEE+XX+YY-U,32:RETURN
2000 FORT=240TO140STEP-1:POKE36876,T:NEXT:POKE36869,240:PRIN
T"▯":POKE36877,0
2005 PRINT"▯▯▯▯DAS SPIEL IST AUS":IFD=0GOTO2020
2010 FORT=140TO240:POKE36876,T:NEXT:PRINT"▯▯BRAVO!!":GOTO203
0
2020 PRINT"▯▯VERSAGER!!"
2030 PRINT"▯▯PUNKTE:"P+100:POKE36876,0:POKE198,0:PRINT"▯▯▯▯▯
NOCH EINMAL?"
2035 GETA$:IFA$=""GOTO2035
```



```
2040 IFA$="J"THENRUN
2050 END
60000 DATA"5  1111111111111111 5","  1             1 "
60005 DATA"   4  11111111111  4  ","  1  4          4  1  ",
"   1 11 1114111 11 1  "
60010 DATA"   1 1            1 1  ","  1    11411141l    1  ",
"   1 1 1       1 1 1  "
60015 DATA"   1 1 4 11411 4 1 1  ","  1 1 1 1   1 1 1 1  ",
"   1 4 1 4 6 4 1 4 1  "
60020 DATA47,63,72,82,99,141,145,183,187,191,225,229,233,237
,271,275,279,317,321,363
60025 DATA380,390,399,415
```

Meteorit

für die Grundversion des VC-20. Mit einer Laserkanone müssen Sie herabfallende Meteoriten abwehren.

Meteoriten, die Ihrer Stadt und Ihrer Kanone mit der Vernichtung drohen. Die Stadt kann jedoch gerettet werden, wenn Sie geschickt und reaktionsschnell Ihre Laserkanone bedienen.
Wird die Kanone getroffen, stehen Ihnen drei weitere Abwehrwaffen zur Verfügung. Nach der Zerstörung von insgesamt vier Kanonen ist das Spiel jedoch beendet. Bei Spielende wird die Anzahl der getroffenen Meteoriten sowie der Verbrauch von Laserkanonen angegeben.
Übrigens: Nach einer gewissen Übungszeit kann die Spielstärke verändert werden. Mit der Eingabe der Spielstärke bestimmt man dann die erlaubte Anzahl von Meteoriteneinschlägen auf die Grundmauern eines Hauses. Gesteuert wird mit dem Joystick. Mit dem Feuerknopf werden die Energiestöße abgegeben.

```
0 POKE36879,237:PRINTCHR$(147):POKE56,28:POKE52,28:CLR
2 PRINT"   ----▪▪METEORIT▪▪----"
4 PRINTTAB(22)" BY ROBERT DIEPOLD"
6 REM 6730 NEUSTADT
8 PRINT"▪▪▪ ▪▪WELCHE SPIELSTAERKE▪▪":PRINT
10 INPUT" ▪1 BIS 10▪";A%
12 IFA%<1ORA%>10THEN10
14 PRINTCHR$(147)
16 FORI=0TO95:POKE7168+I,PEEK(32768+I):NEXTI
18 POKE36869,255
20 FORT=0TO95:READA:POKE7168+T,A:NEXT
22 DATA0,0,0,0,0,0,0,0
24 DATA68,0,10,64,24,60,124,56
26 DATA102,102,255,129,126,60,36,195
28 DATA0,0,102,102,102,0,0,165
30 DATA129,165,165,165,129,165,165,129
32 DATA24,24,24,36,66,129,165,165
34 DATA 0,195,165,153,24,0,0,0
36 DATA129,165,165,129,189,189,189,255
38 DATA255,129,165,165,129,165,165,129
40 DATA254,16,121,159,244,32,160,120
42 DATA0,0,0,0,14,113,128,255
44 DATA0,0,12,106,145,1,1,254
46 FORZ=0TO480:POKE7680+Z,0:NEXTZ
48 PRINT"▪JK@@@JK@@@@JK@@@JK▪@@▪F▪"
50 PRINT:POKE36878,15:L=36874:L1=36875:L2=36876:L3=36877
52 PRINT"▪▪B@B@B▪@@@@@@@▪F@@@@@F▪"
54 PRINT
56 PRINT"▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪E▪H▪E▪▪▪H▪▪▪HHH▪E"
58 PRINT"▪▪IHD▪D▪D▪▪I▪▪D▪▪I▪▪DDD▪D▪I▪"
60 PRINT"EHDD▪D▪DHH▪DHH▪DDDHDH"
62 PRINT"DDDDHDHDDDHDDDHDDDDDD"
64 POKE8163,5:POKE38883,0:FORI=0TO21:POKE8164+I,7:POKE38884+I,0:NEXTI
66 FORO=0TO20:FORI=220TO252:POKEL,I:NEXTI:POKEL,0:NEXTO
68 B1=22:X%=0
70 IFB%=A%THEN192
72 POKEN3,0:X=INT(RND(1)*22)+0:N1=8065:N2=38785:N3=7768+X:N4=38488+X:POKEL3,255
74 BB=22:N9=0:FORI1=0TO5:NEXTI1:POKEL3,0
76 POKEN1+Y,2:POKEN2+Y,6
78 IFY<>0THENN3=N3+BB:N4=N4+BB:GOTO136
80 IFFR=1THENN3=N3+BB:N4=N4+BB:GOTO134
82 POKEN3,0
```



```
84 N3=N3+BB:N4=N4+BB
86 DD=37154:P1=37151:P2=37152:POKEDD,127:P=PEEK(P2)AND128:J0=-(P=0):POKEDD,255
88 P=PEEK(P1):J2=-((PAND16)=0):FR=((PAND32)=0)
90 IFFR=-1THENFR=1
92 IFFR=1THENXX=1
94 IFJ2=1THEN:POKEN1+Y,0:Y=Y-1
96 IFJ0=1THEN:POKEN1+Y,0:Y=Y+1
98 IFY=-12THENY=Y+1
100 IFY=11THENY=Y-1
102 POKEN3,1:POKEN4,2:POKEL3,253
104 N9=N9+1
106 IFN9=17THEN:POKEN3+BB,0:BX=BX+1:GOTO70
108 IFPEEK(N3+BB)=2THEN114
110 IFPEEK(N3+BB)=>4ANDPEEK(N3+BB)=<9THENPOKEN3+BB,0:POKEN3,0:GOTO70
112 GOTO132
114 MU=MU+1
116 IFMU=1THEN:POKE7747,0:GOTO124
118 IFMU=2THEN:POKE7749,0:GOTO124
120 IFMU=3THEN:POKE7751,0:GOTO124
122 IFMU=4THENPOKEL3,0:GOTO188
124 FORUI=0TO5
126 FORIU=254TO128STEP-21:POKE36876,IU:POKE36878,15:NEXTIU:POKE36876,0
128 NEXTUI
130 GOTO70
132 GOTO76
134 N5=N1:N6=N2
136 VC=VC+1
138 N5=N5-B1:N6=N6-B1
140 POKEN5+Y,3:POKEN6+Y,2:POKEL3,0:POKEL,129
142 IFPEEK(N5+Y-66)=1THEN:POKEN3,0:GOTO164
144 IFPEEK(N5+Y-22)=1THEN:POKEN3,0:GOTO164
146 IFPEEK(N5+Y-88)=1THEN:POKEN3,0:GOTO164
148 IFPEEK(N5+Y-110)=1THEN:POKEN3,0:GOTO164
150 IFPEEK(N5+Y-132)=1THEN:POKEN3,0:GOTO164
152 IFPEEK(N5+Y-44)=1THEN:POKEN3,0:GOTO164
154 POKEN5+Y,0:POKEL,0:POKEL3,254
156 IFVC=3THENVC=0:POKEN3-BB,0:GOTO86
158 FR=-1
160 POKEN3-BB,0
162 GOTO102
164 POKE36874,0:E=E+1
166 POKEN5+Y,0:POKEN3-BB,0
168 FORNN=0TO100
170 POKEL3,130
172 NEXTNN
174 POKEL3,0
176 FORCV=0TO21:POKEL3,235
178 POKE7701-CV,9:POKE38421-CV,2
180 FORQ5=0TO12:NEXTQ5
182 POKE7701-CV,0:POKEL3,0
184 NEXTCV
186 POKEL3,0:CV=0:VC=0:FR=0 :GOTO68
188 PRINT"▯":POKE36869,240:POKE36879,238:FORPK=254TO127STEP-1:POKEL2,PK:NEXTPK
190 PRINT" ▯▯▯  DU HAST DEINE VIER     LASERKANONEN               VERBRAUCHT.":GOTO194
192 POKE36869,240:POKE36879,122:PRINTCHR$(147):POKEL3,0
194 PRINT"▯DU HAST ";E
196 PRINT"▯METEORITEN ZERSTOERT!"
198 PRINT"EIN NEUES SPIEL??"
200 PRINT" ▯J/N▯"
202 GETA$:IFA$="J"THEN0
204 IFA$="N"THENEND
206 GOTO202
```

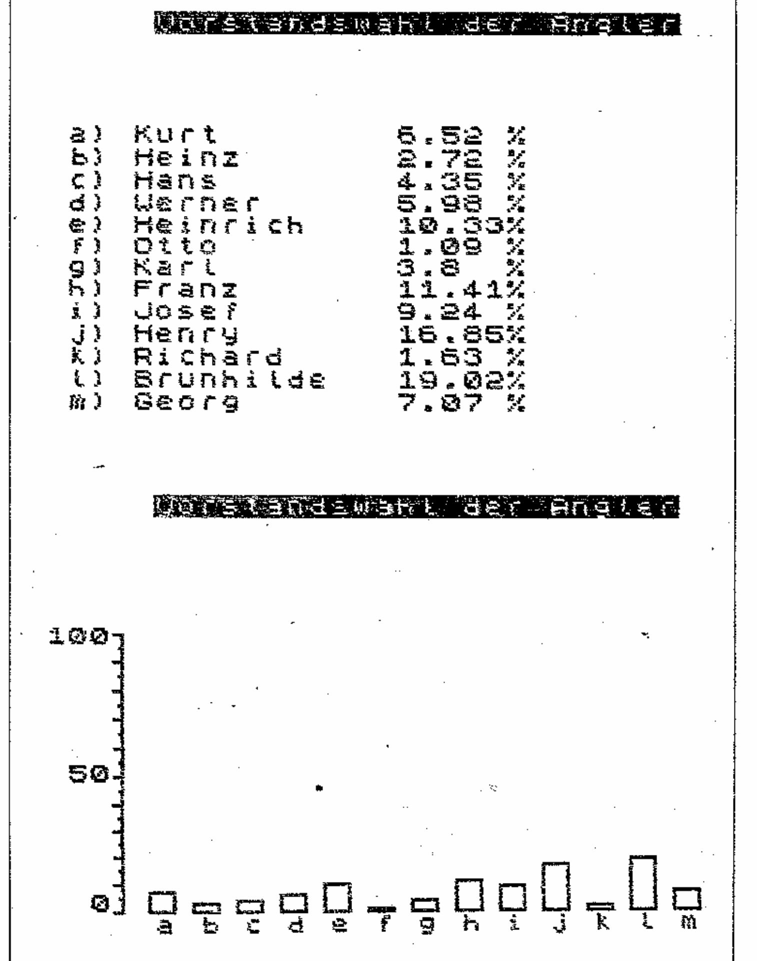
Superstat



Superstat ist ein Statistikprogramm. Es errechnet Prozentanteile und erstellt Balkengrafiken für max. 14 frei benennbare Punkte. Zuerst erfolgt eine Bildschirmgrafik, die auf Wunsch ausgedruckt werden kann, dann eine genaue Angabe der erreichten Prozente.

Ist kein Drucker vorhanden, müssen die Zeilen 610 und 740 sowie in den jeweiligen Vorzeilen die entsprechenden Angaben weggelassen werden.

```
  10 REM SUPERSTAT von H.Stoehr
  20 REM am 8.4.1984
  30 REM ©BANANASOFT
  40 REM
  50 REM Speicherbedarf:1,93 K
  55 REM
  90 REM Eingabe
  95 PRINT AT 11,11; INK 2; FLAS
H 1;"EINGABE"; FLASH 0
  96 INK 0
  98 INPUT "Ueberschrift (max. 2
5 Buchst.) :";u$
 100 INPUT "Anzahl (max. 14): ";
anz
 102 IF anz>14 THEN LET anz=14
 110 DIM n$(anz,10): DIM w(anz):
 DIM p(anz): DIM i(anz): DIM b$(
anz,1)
 111 FOR n=1 TO anz
 112 READ b$(n)
 115 INPUT ("Name ";n;"(max. 10
Buchst.):  ");n$(n)
 116 IF LEN n$(n)>10 THEN LET n$
(n)=n$(n)( TO 10)
 120 NEXT n
 130 INPUT "Art: ";a$
 140 FOR n=1 TO anz
 150 INPUT ("Wieviel ";a$;" erhi
elt ";n$(n);" ?");w(n)
 155 NEXT n
 195 REM BERECHNUNGEN
 200 LET s=w(1)
 210 FOR n=2 TO anz
 220 LET s=s+w(n): NEXT n
 230 LET pro=s/100
 240 FOR n=1 TO anz
 250 LET p(n)=(INT ((w(n)/pro)*1
00+.5)/100)
 260 LET i(n)=INT (w(n)/pro+.5)
 270 NEXT n
 290 REM MASSTAB
 295 CLS
 300 INK 1
 310 PLOT 28,16: DRAW 0,100
 320 FOR m=21 TO 111 STEP 10
 330 FOR n=16 TO 116 STEP 10
 340 PLOT 25,n: DRAW 3,0
 350 NEXT n
 360 PLOT 27,m: DRAW 1,0
 370 NEXT m
 380 PRINT AT 19,2;"0";AT 13,1;"
50";AT 7,0;"100"
 390 REM GRAFIK
 400 PRINT AT 1,5; INVERSE 1;u$;
 INVERSE 0
 410 LET x=38
 420 FOR i=1 TO anz
 430 PLOT x,16: DRAW 9,0: DRAW 0
,i(i)
 440 PLOT x,16: DRAW 0,i(i): DRA
W 9,0
 450 LET x=x+16
 460 NEXT i
 470 LET m=5
 480 FOR n=1 TO anz
 490 INK 2: PRINT AT 20,m;b$(n)
 500 LET m=m+2
 510 NEXT n
 520 INK 0
 530 PRINT #1;"AUSDRUCK:       'A
'"
 540 PRINT #0;"GENAUE ZAHLEN:  'Z
'"
 600 PAUSE 0: LET z$=INKEY$
 610 IF z$="a" THEN COPY
 620 IF z$="z" THEN GO TO 640
 625 GO TO 600
 630 REM GENAUE ZAHLEN
 640 CLS
 650 PRINT AT 1,5; INVERSE 1;u$;
 INVERSE 0
 655 PRINT : PRINT : PRINT : PRI
NT
 660 FOR n=1 TO anz
 670 PRINT " "; INK 2;b$(n);") "
; INK 0;n$(n);"  ";p(n);TAB 21;"
%"
 680 NEXT n
 700 PRINT #1;"AUSDRUCK:  'A'"
 710 PRINT #0;"GRAFIK:    'G'
ENDE:  'E'"
 720 PAUSE 0: LET x$=INKEY$
 730 IF x$="g" THEN GO TO 290
 740 IF x$="a" THEN COPY
 750 IF x$="e" THEN STOP
 770 GO TO 720
9000 DATA "a","b","c","d","e","f
","g","h","i","j","k","l","m","n
"
```





„2fache Vergrößerung"

Für ZX-Spectrum 16/48 K!
Wie kann eine 2fache Vergrößerung von ein oder mehr Zeichen und Grafikzeichen erreicht werden? 2fach vergrößerte Zeichen lassen sich z. B. in Spielen oder zu Anzeigezwecken verwenden.

Programmbeschreibung:
In Zeile 90 wird die Größe des/der zu berechnenden Zeichen in einem Feld gespeichert. Durch INK 7 in Zeile 100 ist das/die in Zeile 110 geprinteten Zeichen nicht zu sehen und wird ab Zeile 120 in das Feld „a" eingelesen.
Bei diesem Vorgang könnte es auch, ohne daß es vorher in ein Feld abgespeichert wird, gleich geplottet werden. Ab Zeile 190 erfolgt der Ausdruck. In Zeile 210 kann die Ausdruckstelle geändert werden. Das Programm sollte speziell modifiziert in andere Programme eingebaut werden.
Durch das einfache Grundprinzip läßt sich das Programm sicher auch auf andere Computer übertragen.

```
 10 REM 2-FACHE VERGROESSERUNG
 20 REM
 30 REM  by Thomas Schroeer
 40 REM       Tel.06383-7490
 50 REM
 60 REM
 70 REM Berechnung + Speichern
 80 REM im Feld 'a'
 90 DIM a(8,8)
100 INK 7
110 PRINT AT 21,0;"1": REM Zu vergroesserndes Zeichen
120 FOR b=0 TO 7
130 FOR c=0 TO 7
140 IF POINT (c,b) THEN LET a(c,b)=1
150 NEXT c
160 NEXT b
170 INK 0
180 REM Ausdruck
190 FOR b=1 TO 8
200 FOR c=1 TO 8
210 IF a(c,b)=1 THEN PLOT 80+2*c,b*2: PLOT 79+2*c,b*2: PLOT 80+2*c,b*2+1: PLOT
79+2*c,b*2+1
220 NEXT c
230 NEXT b
```

>Kleinstes gemeinsames Vielfaches<

Dieses Programm ist ein Hilfsprogramm für Mathematik, daß z. B. das Erweitern von Brüchen erleichtert. Für den ZX-Spectrum 16/48 K

Programmbeschreibung:
In den Zeilen 120 und 130 wird der Variable „g" die größte und der Variable k die kleinste der beiden eingegebenen Zahlen zugeteilt. In Zeile 140 können nach Bedarf weitere 9er eingefügt werden, um den Berechnungsbereich zu erhöhen. In dieser Zeile wird in Schritten der größten Zahl gezählt, bis in Zeile 150 eine ganzzahlige Zahl herauskommt. In Zeile 170 wird das KGV ausgegeben, gefolgt von einer schönen Tonfolge.
Auf andere Computer läßt sich das Programm sehr leicht umschreiben.

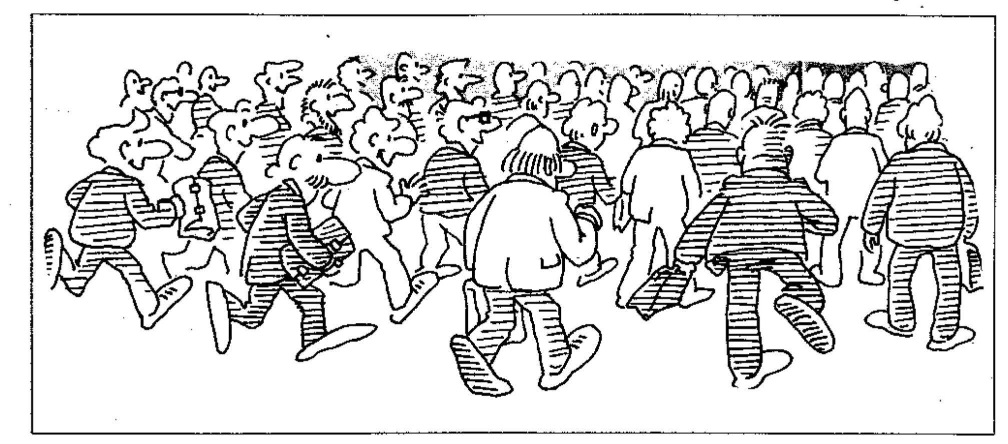
```
 10 REM KLEINSTES GEMEINSAMES
 20 REM VIELFACHES
 30 REM
 40 REM  by Thomas Schroeer
 50 REM Tel.06383-7490
 60 REM
 70 REM
 80 CLS : BEEP .1,0
 90 PRINT "KLEINSTES GEMEINSAMES VIELFACHES================================"
100 INPUT "1.Zahl: ";a
110 INPUT "2.Zahl: ";b
120 IF a>b THEN LET g=a: LET k=b: GO TO 140
130 LET g=b: LET k=a
140 FOR n=g TO 9999 STEP g
150 IF n/k=INT (n/k) THEN GO TO 170
160 NEXT n
170 PRINT AT 10,1;"1.Zahl: ";a'" 2.Zahl: ";b''" KGV: ";n;AT 21,0;"TASTE !!!"
180 BEEP .5,8: BEEP .5,6: BEEP 1,4
190 PAUSE 0
200 GO TO 80
```

Irrgarten 3D



für ZX - 81, 16k

Am Anfang muß Teil 1 des Programmes eingegeben werden. Mit RUN wird es gestartet und auf dem Bildschirm erscheinen die möglichen 8 Bilder. Danach darf unter keinen Umständen mehr RUN benutzt werden, nur noch GOTO!!! Zur Sicherheit sollte das Programm abgespeichert werden. Nun kann das Hauptprogramm eingegeben werden. **Sinn des Spieles ist es, mit möglichst wenigen Schritten Deinen Freund aus dem Irrgarten zu leiten.** Bewahre die Übersicht und leite ihn sicher zum Ausgang! Damit es für Dich nicht zu schwer wird, kannst Du bis zu zehn mal pro Spiel eine Übersicht auf einem 20 × 20 Schritt großen Feld abrufen. Bei der Eingabe "A" läuft Dein Freund automatisch weiter, wenn es nur eine Möglichkeit gibt. Gibst Du "W" ein, wartet er auf Deine Anweisungen. Es ist manchmal ganz schön verhext, aber keine Angst, Du wirst den Ausgang schon finden!



Hier noch ein Hinweis für Computer-Fans, die einen Q-save besitzen: Das Programm erst ganz eingeben, abspeichern und dann erst laufen lassen, sonst wird die Software für den Q-save zerstört!

TEIL 1

```
10 DIM W$(8,768)
20 LET A$="███████████"
30 LET B$="▒▒▒▒▒▒▒▒▒▒▒▒"
40 LET C$="
"
50 POKE 16418,0
60 FOR Q=1 TO 8
70 SLOW
80 PRINT AT 0,0;"▒▒▒▒";AT 0,28;"▒▒▒▒";AT 23,0;"▒▒▒▒";AT 23,28;"▒▒▒▒"
90 FOR W=1 TO 11
100 PRINT AT W,0;"▒▒▒"+A$( TO W)+"▒"+C$( TO 24-2*W)+"▒"+A$( TO W)+"▒▒▒"
110 PRINT AT 23-W,0;"▒▒▒"+A$( TO W)+"▒"+C$( TO 24-2*W)+"▒"+A$( TO W)+"▒▒▒"
120 NEXT W
130 PRINT AT 11,14;"▒▒▒▒";AT 12,14;"▒▒▒▒"
140 IF Q=5 THEN GOTO 250
150 FOR W=3 TO 20
160 IF Q=1 OR Q=2 OR Q=6 THEN GOTO 180
170 PRINT AT W,6;(C$( TO 4) AND (W<7 OR W>16))+(B$( TO 4) AND (W>6 AND W<17))
180 IF Q=1 OR Q=3 OR Q=7 THEN GOTO 200
190 PRINT AT W,22;(C$( TO 4) AND (W<7 OR W>16))+(B$( TO 4) AND (W>6 AND W<17))
200 NEXT W
210 IF Q>5 THEN GOTO 250
220 FOR E=7 TO 16
230 PRINT AT E,10;B$( TO 12)
240 NEXT E
250 FAST
260 FOR W=0 TO 23
270 FOR E=1 TO 32
280 LET W$(Q,E+32*W)=CHR$ PEEK (PEEK 16396+256*PEEK 16397+E+33*W)
290 NEXT E
300 NEXT W
310 NEXT Q
320 SAVE "I"
330 LET B$=""
340 LET C$=""
350 STOP
```

TEIL 2

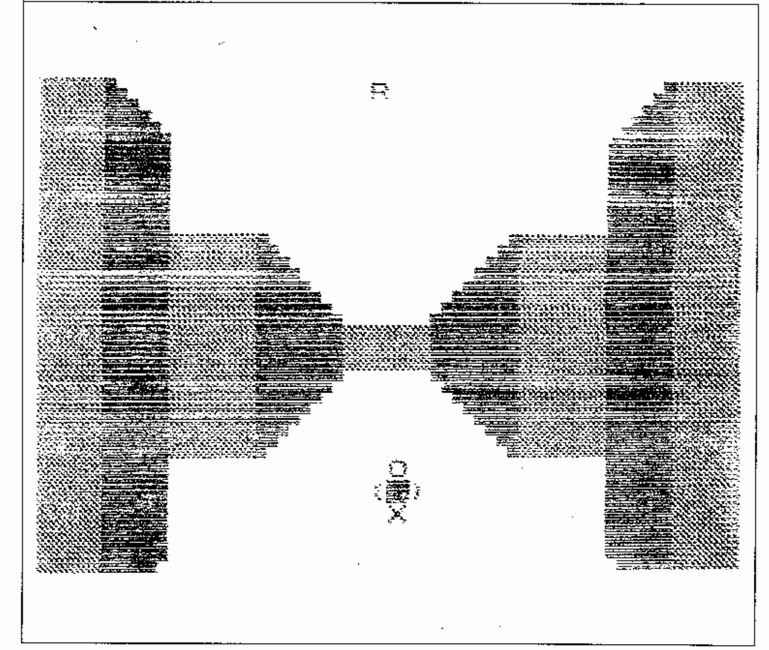
```
10 CLS
20 PRINT AT VAL "5",VAL "5";"I R R G A R T E N"
30 PRINT ,,"HELFE MIR BITTE MIT MOEGLICHST  WENIGEN SCHRITTEN AUS DIESEN           IRRGARTEN."
40 PRINT ,,"DU KANNST ZEHNMAL EINE UEBER-    SICHT BEKOMMEN, WENN DU AUF      ""H"" DRUECKST."
50 PRINT ,,"SOLL ICH VON SELBST WEITER        GEHEN,WENN ES NUR EINE MOEG-     LICHKEIT GIBT?"
60 PRINT ,,"▓UTOMATISCH WEITER LAUFEN."
70 PRINT ,,"▓ARTEN AUF DIE EINGABE."
80 IF INKEY$="" THEN GOTO VAL "80"
90 LET B$=INKEY$
100 IF B$<>"A" AND B$<>"W" THEN GOTO VAL "80"
110 CLS
120 PRINT AT VAL "5",VAL "0";"BITTE ETWA 4 MINUTEN WARTEN, DERGARTEN WIRD GERADE AUFGEBAUT."
130 PRINT ,,,,"STEUERUNG:","""7""  ",,"""5""  *  ""8""",,"""6"""
140 DIM I$(VAL "70",VAL "70")
150 FOR Q=VAL "1" TO VAL "70"
160 LET I$(VAL "1",Q)="▒"
170 LET I$(Q,VAL "1")="▒"
180 LET I$(Q,VAL "70")="▒"
190 LET I$(VAL "70",Q)="▒"
200 NEXT Q
210 LET X=VAL "35"
220 LET Y=X
230 FAST
240 FOR Q=VAL "1" TO VAL "9"
250 IF Q=VAL "1" THEN GOTO VAL "290"
260 LET X=INT (RND*VAL "50"+VAL "5")
270 LET Y=INT (RND*VAL "50"+VAL "5")
280 IF I$(X+VAL "1",Y)="▒" OR I$(X-VAL "1",Y)="▒" OR I$(X,Y+VAL "1")="▒" OR I$(X,Y-VAL "1")="▒" THEN GOTO VAL "260"
290 LET X1=(RND<VAL ".5")*SGN (RND-VAL ".5")
300 FOR W=VAL "1" TO INT (RND*VAL "5")+VAL "8"
310 LET Y1=(X1<>VAL "0")*SGN (RND-VAL ".5")
320 LET X1=(X1=VAL "0")*SGN (RND-VAL ".5")
330 LET X1=X1*(Y1=VAL "0")
340 LET Y1=Y1*(X1=VAL "0")
```



```
 350 FOR E=VAL "1" TO INT (RND*V
AL "2")+VAL "2"
 360 LET I$(X,Y)="▒"
 380 IF Y1=VAL "0" THEN IF I$(X+
X1,Y-VAL "1")="▒" OR I$(X+X1,Y+V
AL "1")="▒" THEN GOTO VAL "420"
 400 IF X1=VAL "0" THEN IF I$(X+
VAL "1",Y+Y1)="▒" OR I$(X-VAL "1
",Y+Y1)="▒" THEN GOTO VAL "420"
 410 GOTO VAL "440"
 420 IF I$(X+X1,Y+Y1)="▒" THEN G
OTO VAL "440"
 430 GOTO VAL "490"
 440 LET X=X+X1
 450 LET Y=Y+Y1
 460 IF Q=VAL "1" THEN IF I$(X+X
1,Y+Y1)="▒" THEN GOTO VAL "530"
 470 IF I$(X,Y)="▒" THEN GOTO VA
L "260"
 480 NEXT E
 490 NEXT W
 500 IF Q=VAL "1" THEN GOTO VAL
"310"
 510 NEXT Q
 520 GOTO VAL "560"
 530 LET I$(X,Y)="▒"
 540 LET I$(X+X1,Y+Y1)="A"
 550 GOTO VAL "510"
 570 LET X=VAL "35"
 580 LET Y=X
 590 LET N=VAL "0"
 600 LET H=VAL "10"
 610 FOR S=VAL "0" TO VAL "3"
 620 LET X1=(S=VAL "2")-(S=VAL "
0")
 630 LET Y1=(S=VAL "3")-(S=VAL "
1")
 640 IF I$(X+X1,Y+Y1)="▒" THEN G
OTO VAL "680"
 650 NEXT S
 660 LET X=X+X1
 670 LET Y=Y+Y1
 700 LET W=VAL "1"
 710 IF I$(X-Y1,Y+X1)="▒" THEN L
ET W=W+VAL "2"
 720 IF I$(X+Y1,Y-X1)="▒" THEN L
ET W=W+VAL "1"
 730 IF I$(X,Y)="A" THEN GOTO VA
L "3000"
 740 IF I$(X+X1,Y+Y1)="A" OR I$(
X+X1,Y+Y1)="▒" THEN LET W=W+VAL
"4"
 790 POKE VAL "16418",VAL "0"
 800 LET N=N+VAL "1"
 810 CLS
 820 SLOW
 830 PRINT AT VAL "0",VAL "0";W$
(W)
 840 PRINT AT VAL "0",VAL "15";(
"O" AND S=VAL "0");("R" AND S=VA
L "1");("U" AND S=VAL "2");("L"
AND S=VAL "3")
 850 PRINT AT VAL "21",VAL "16";
"O";TAB VAL "15";"(▒)";TAB VAL "
16";"X"
 860 FOR Q=VAL "20" TO VAL "17"
STEP VAL "-1"
 870 PRINT AT Q,VAL "16";"O";TAB
 VAL "15";"(▒)";TAB VAL "15";" X
";TAB 16;" "
 880 NEXT Q
 890 LET A$=INKEY$
 900 IF B$="A" AND A$="" THEN LE
T A$=("7" AND W=VAL "5")+("8" AN
D W=VAL "2")+("5" AND W=VAL "3")
 910 IF A$="" AND INKEY$="" THEN
 GOTO VAL "910"
 920 IF A$="" THEN LET A$=INKEY$
 930 IF A$="W" OR A$="A" THEN LE
T B$=A$
 940 LET S=S+(A$="8")-(A$="5")+V
AL "2"*(A$="6")
 950 IF S<VAL "0" THEN LET S=S+V
AL "4"
 960 IF S>VAL "3" THEN LET S=S-V
AL "4"
 970 LET X1=(S=VAL "2")-(S=VAL "
0")
 980 LET Y1=(S=VAL "1")-(S=VAL "
3")
 990 IF A$="H" THEN GOTO VAL "20
00"
1000 IF A$="6" THEN GOTO VAL "13
00"
1010 IF E=5 AND (A$="O" OR A$="P
") THEN GOTO VAL "1050"
1020 IF A$="5" THEN GOTO VAL "11
10"
1030 IF A$="8" THEN GOTO VAL "12
00"
1040 IF A$<>"7" THEN GOTO VAL "8
90"
1050 IF W<VAL "5" THEN GOTO VAL
"890"
1060 FOR Q=VAL "16" TO VAL "14"
STEP VAL "-1"
1070 PRINT AT Q,VAL "16";"O";TAB
 VAL "15";"(▒)";TAB VAL "15";" X
";TAB 16;" "
1080 NEXT Q
1090 PRINT AT Q+VAL "2",VAL "15"
;"   ";TAB VAL "16";" ";AT Q,VAL
 "16";" ";TAB VAL "16";"A";AT Q,
VAL "16";" ";TAB VAL "16";" "
1100 GOTO VAL "660"
1110 PRINT AT VAL "18",VAL "15";
" ▒ ";TAB VAL "16";"J"
1120 IF W<>VAL "4" AND W<>VAL "3
" AND W<>VAL "8" AND W<>VAL "7"
THEN GOTO VAL "890"
1130 FOR Q=VAL "16" TO VAL "6" S
TEP VAL "-1"
1140 PRINT AT VAL "17",Q;"O ";TA
B Q;"▒ ";TAB Q;"J "
1150 NEXT Q
1160 PRINT AT VAL "17",VAL "6";"
 ";TAB VAL "6";" ";TAB VAL "6";"
 "
1170 GOTO VAL "660"
1200 PRINT AT VAL "18",VAL "15";
" ▒ ",TAB VAL "16";"L"
1210 IF W<>VAL "2" AND W<>VAL "4
" AND W<>VAL "8" AND W<>VAL "6"
THEN GOTO VAL "890"
1220 FOR Q=VAL "16" TO VAL "24"
1230 PRINT AT VAL "17",Q;" O";TA
B Q;" ▒";TAB Q;" L"
1240 NEXT Q
1250 PRINT AT VAL "17",Q;" ";TAB
 Q;" ";TAB Q;" "
1260 GOTO VAL "660"
1300 FOR Q=VAL "18" TO VAL "21"
1310 PRINT AT Q-VAL "1",VAL "15"
;"   ";AT Q,VAL "15";" O ";AT Q+
VAL "1",VAL "15";"(▒)",TAB VAL "
15";"X"
1320 NEXT Q
1330 PRINT AT Q-VAL "1",VAL "16"
;" ";TAB VAL "15";"   ",TAB VAL
"16";" "
1340 GOTO VAL "660"
2000 IF H=VAL "0" THEN GOTO VAL
"890"
2010 LET H=H-VAL "1"
2020 LET EY=(Y-VAL "55")*((Y+VAL
 "15")<VAL "70")+VAL "70"
2030 LET EX=(X-VAL "60")*((X+VAL
 "10")<VAL "70")+VAL "70"
2040 LET AY=(Y-VAL "10")*((Y-VAL
 "10")>VAL "0")+VAL "1"
2050 LET AX=(X-VAL "10")*((X-VAL
 "10")>VAL "0")+VAL "1"
2060 LET I$(X,Y)="▒"
2070 CLS
2080 PRINT AT VAL "2",VAL "6";
2090 FOR R=AX TO EX
2100 PRINT TAB VAL "6";I$(R,AY T
O EY)
```



```
2110 NEXT R
2120 LET I$(X,Y)="▒"
2130 PRINT AT VAL "0",VAL "10";"
NOCH ";H;"MAL"
2140 IF INKEY$="" THEN GOTO VAL
"2140"
2150 LET Q=Q+VAL "1"
2160 PRINT AT VAL "0",VAL "0";W$
(Q)
2170 GOTO 870
3000 PRINT AT VAL "9",VAL "9";"A
 U S G A N G"
3010 FOR Q=VAL "1" TO VAL "99"
3020 NEXT Q
3030 CLS
3040 PRINT AT VAL "5",VAL "0";"D
U MUSSTEST ";N;" SCHRITTE GEHEN"
;AT VAL "9",VAL "9";"NOCH EIN SP
IEL ?"
3050 PRINT AT VAL "15",VAL "15";
"▒";"X"
3060 PRINT AT VAL "14",VAL "16";
"O/";AT VAL "14",VAL "17";")"
3070 IF INKEY$="J" THEN GOTO 1
3080 IF INKEY$="N" THEN STOP
3090 GOTO 3050
9000 DIM I$(1,1)
9005 SAVE "I"RRGARTEN 3D
9010 GOTO 1
```

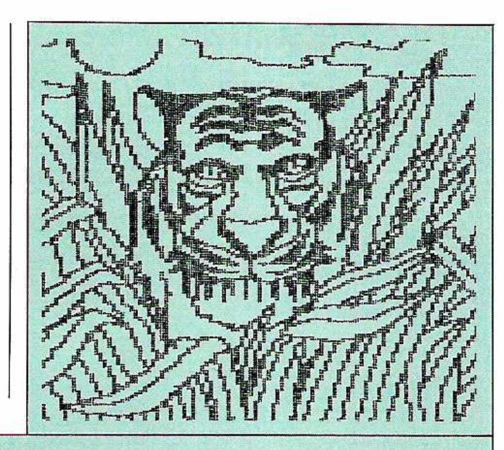


DRAGON 32 THE FAMILY COMPUTER

Hardcopy V1.1

Das hier vorgestellte Programm "Hardcopy" für den Dragon 32 oder 64 erzeugt einen Bildschirmausdruck auf dem Seikosha GP-100 A Drucker oder einem ähnlichen Modell. Das Programm ist in positions-unabhängigem Maschinencode geschrieben (PIC). Da ein Punkt-für-Punkt-Ausdruck in einem sehr kleinen Bild resultiert, wurde eine etwas andere Form gewählt. Das Bild wird um 90 Grad gedreht und zweifach vergrößert auf dem Drucker ausgegeben. Eingegeben werden kann Hardcopy mit dem abgedruckten Lade-Programm. Eine wichtige Anmerkung muß noch gemacht werden: durch POKEn in Adresse 7697 mit 0 oder 255 wird bestimmt, ob das Bild original oder invertiert ausgedruckt werden soll. Der Ausdruck selbst dauert ungefähr 2-3 Minuten.

Hinweis:
Vor dem Start des Programms mit EXEC 7741 sollte es abgespeichert werden mit: CSAVEM"HARDCOPY",7697,8848,7741.



```
5 PCLEAR5
10 FOR A=7697 TO 8147 STEP 10
20 C=0
30 FOR B=0 TO 9
40 PRINT A+B;"=";:INPUT A$
50 POKE A+B,VAL("&H"+A$)
60 C=C+VAL("&H"+A$)
70 NEXT
80 INPUT"PRUEFSUMME";P
90 IF P<>C THEN B=0:PRINT"FEHLER!ZEILE
NEU EINGEBEN!!":GOTO20 ELSE NEXT
```

```
 7697 BIS 7706
 FF   34   12   BE   1
 6B   AF   8C   E9   30
Pruefsumme: 1219
 7707 BIS 7716
 8D   0    D    BF   1
 6B   86   7E   B7   1
Pruefsumme: 897
 7717 BIS 7726
 7F   A7   8C   D9   35
 92   D    6F   26   D3
Pruefsumme: 1223
 7727 BIS 7736
 34   2    96   B6   81
 4    35   2    26   C9
Pruefsumme: 813
 7737 BIS 7746
 81   15   26   C5   86
 FF   A7   8C   CE   6F
Pruefsumme: 1398
 7747 BIS 7756
 8C   C9   86   80   A7
 8C   C1   6F   8C   BC
Pruefsumme: 1542
 7757 BIS 7766
 86   1    A7   8C   BD
 86   FE   97   6F   86
Pruefsumme: 1415
 7767 BIS 7776
 8    AD   9F   A0   2
 9E   BA   30   88   1F
Pruefsumme: 1061
 7777 BIS 7786
 AF   8C   A1   30   89
 17   E0   AF   8C   9C
Pruefsumme: 1379
 7787 BIS 7796
 10   8E   0    1    8D
 6C   AE   8C   91   17
Pruefsumme: 890
 7797 BIS 7806
 0    88   17   1    14
 17   0    95   30   88
Pruefsumme: 536
 7807 BIS 7816
 20   AC   8C   84   2F
 EF   86   D    AD   9F
Pruefsumme: 1241
 7817 BIS 7826
 A0   2    8D   50   AE
 8D   FF   74   9C   BA
Pruefsumme: 1411
 7827 BIS 7836
 2D   5B   8D   68   17
 1    33   8D   76   30
Pruefsumme: 763
 7837 BIS 7846
 88   20   AC   8D   FF
 64   2F   F0   86   D
Pruefsumme: 1270
 7847 BIS 7856
 AD   9F   A0   2    AE
 8D   FF   56   9C   BA
Pruefsumme: 1492
 7857 BIS 7866
 2D   45   31   21   10
 8C   0    8    2F   C
Pruefsumme: 419
 7867 BIS 7876
 10   8E   0    1    6C
 8D   FF   43   6C   8D
Pruefsumme: 979
 7877 BIS 7886
 FF   41   AE   8D   FF
 3A   30   1F   AF   8D
Pruefsumme: 1343
 7887 BIS 7896
 FF   34   AE   8D   FF
 32   30   1F   AF   8D
Pruefsumme: 1322
 7897 BIS 7906
 FF   2C   20   92   86
 1C   AD   9F   A0   2
Pruefsumme: 1133
 7907 BIS 7916
 86   2F   AD   9F   A0
 2    86   80   AD   9F
Pruefsumme: 1269
```




```
 7917 BIS 7926
  A0   2    39   86   81
  A7   8D   FF   1A   20
Pruefsumme: 1103
 7927 BIS 7936
  9D   86   1E   AD   9F
  A0   2    39   1F   20
Pruefsumme: 935
 7937 BIS 7946
  E7   8D   FF   5    EC
  84   6A   8D   FE   FF
Pruefsumme: 1756
 7947 BIS 7956
  27   4    59   49   20
  F6   39   A8   8D   FE
Pruefsumme: 1103
 7957 BIS 7966
  FB   8A   80   A4   8D
  FE   F4   A7   8D   FE
Pruefsumme: 1882
 7967 BIS 7976
  EC   6C   8D   FE   E5
  A1   8D   FE   E3   26
Pruefsumme: 1789
 7977 BIS 7986
  F    E6   8D   FE   DB
  C1   80   27   7    AC
Pruefsumme: 1398
 7987 BIS 7996
  8D   FE   D1   2C   1
  39   E6   8D   FE   CC
Pruefsumme: 1535
 7997 BIS 8006
  5D   27   40   86   1C
  AD   9F   A0   2    A6
Pruefsumme: 1018
 8007 BIS 8016
  8D   FE   BF   34   2
  AB   E0   AD   9F   A0
Pruefsumme: 1527
 8017 BIS 8026
  2    A6   8D   FE   B5
  AD   9F   A0   2    6F
Pruefsumme: 1349
 8027 BIS 8036
  8D   FE   AB   AC   8D
  FE   A5   2D   13   A6
Pruefsumme: 1528
 8037 BIS 8046
  8D   FE   A4   AD   9F
  A0   2    AD   9F   A0
Pruefsumme: 1545
 8047 BIS 8056
  2    86   80   A7   8D
  FE   95   39   A6   8D
Pruefsumme: 1339
 8057 BIS 8066
  FE   91   A7   8D   FE
  8C   39   A6   8D   FE
Pruefsumme: 1719
 8067 BIS 8076
  87   AD   9F   A0   2
  AD   9F   A0   2    20
Pruefsumme: 1155
 8077 BIS 8086
  D0   A7   8D   FE   7B
  A4   8D   FE   79   E6
Pruefsumme: 1803
 8087 BIS 8096
  8D   FE   75   3D   68
  8D   FE   70   1F   98
Pruefsumme: 1367
 8097 BIS 8106
  58   34   4    AB   E0
  AB   8D   FE   64   A7
Pruefsumme: 1372
 8107 BIS 8116
  8D   FE   60   A6   8D
  FE   5D   81   8    2F
Pruefsumme: 1329
 8117 BIS 8126
  11   6F   8D   FE   55
  6C   8D   FE   51   A6
Pruefsumme: 1358
 8127 BIS 8136
  8D   FE   4C   6F   8D
  FE   48   39   A6   8D
Pruefsumme: 1413
 8137 BIS 8146
  FE   42   20   C1   44
  44   44   8D   BC   44
Pruefsumme: 1146
 8147 BIS 8156
  39   CC   6    0    97
  F5   DD   F6   8B   18
Pruefsumme: 1293
```

Compumask

SCHLUSS mit dem unproduktiven Stöbern in überall herumliegenden Programmierunterlagen.
SCHLUSS mit dem Frust und dem Zeitverlust!
Hunderttausende besitzen bereits einen Heimcomputer, aber nur wenige sind bis zum Programmieren von Sprites und Musik vorgedrungen.
DAS IST AB SOFORT ANDERS!

COMPUmask-Merkmale (mit COMMODORE-Copyright-Genehmigung) Abrieb- und reinigungsmittelfester, veredelter Druck (kein Siebdruck!) mit Schutzschicht auf reißresistenter, flexibler, bis 80°C wärmefester Kunststoff-Platte, nur 0,4 mm stark. Rundfenster für Kontroll-Leuchte. Ober- und Unterseite voll ausgenutzt für die Aufnahme der am häufigsten vorkommenden Daten, Funktionen, Befehle, Zeichen, POKEs, Tabellen usw. (siehe Rückseite). Oberseite 2farbig, wobei die auf den Bildschirm bezogenen Teile zur leichteren Orientierung farbig (hellocker) unterlegt sind.
Die Lieferung erfolgt in PVC-Minigrip-Verschlußtasche mit Kartonverstärkung für Versand und Aufbewahrung. Der durchschnittliche Verkaufspreis liegt bei 29,80 (inkl. Mehrwertsteuer). Der Verkauf erfolgt über den Fachhandel. Folgende weitere COMPUmask sind in Bälde erhältlich: Für VC 20: PEEKs und POKEs, ExBasic Level II. Für C64: SIMON's BASIC, PEEKs und POKEs, ExBasic Level II. Blanko-Masken für VC 20 und C 64, beschreib- und radierfähig, für individuelle und zeitweilige Beschriftung bei der Programmierarbeit. Stapelbar. Lieferung in 5er-Packs. Weiter sind Computermasken in Vorbereitung für eine Reihe anderer Masken und Modelle.

INHALTSVERZEICHNIS
COMPUmask VC 20

1. Mehr als 80 BASIC-Begriffe
2. 20 Fehlermeldungen
3. Zeichen der Rechenoperationen
4. Zeichen der logischen Operationen
5. Bildschirmausgabezeichen der 8 Zeichenfarben im Programm-Modus
6. Bildschirmausgabezeichen der F-Tasten (F1 – F8) im Programm-Modus
7. Bildschirmausgabezeichen der 4 Cursor-Richtungen im Programm-Modus
8. Bildschirmausgabezeichen für CLR/HOME und INST/DEL im Programm-Modus
9. POKEs für Bildschirmrahmen und -Hintergrund
10. POKEs für 3 Stimmen und insgesamt 5 Oktaven
11. POKEs für Bildschirmpositionen und ihre Farben
12. POKEs für die Zeichenfarben
13. Umrechnung BINÄR/DEZIMAL
14. Zeiteingabe
15. Beschreibung der Wirkungsweise wichtiger Funktionstasten
16. 8 wichtige Drucker-Operationen
17. 7 wichtige Floppy-Operationen
18. 71 als Variable ungeeignete Wörter
19. Tafel der Speicherorganisation
20. Bildschirm-Codes
21. ASCII- und CHR$-Codes
22. 3. Demo-Programme für Sound und Grafik
23. BINÄR/DEZIMAL/HEXADEZIMAL-Tabelle

COMPUmask COMMODORE 64, wie VC 20, jedoch:

5. Bildschirmausgabezeichen der 16 Zeichenfarben im Programm-Modus
12. POKEs für 3 Stimmen und Geräusch über je 4 Oktaven
19. entfällt

weiter zusätzlich:

24. POKEs für Hüllkurve, Wellenform, Filter und Notenwerte
25. POKEs für Sprites und Bildschirm-Scrolling
26. Sprite-DATA-Berechnungsschablone
27. Sprite-DEMO-Programm

gesehen bei: Fa. J. Winkler, D-5760 Arnsberg 1

Impressum: Computronic erscheint monatlich im Verlag: Tronic-Verlagsgesellschaft mbH, Landstr. 29, 3444 Wehretal 1, Tel.: (0 56 51) 4 06 43 / 4 06 93. Redaktion: (Verantw.) Rainer Beck, Axel Gredè, Frank Brall. Freie Mitarbeiter: G. Wagner, Volker Bekker, Rolf Freitag. Gesamtherstellung: D + V Paul Dierichs GmbH + Co. KG. Kassel. Vertrieb: Inland (Groß-, Einzel- und Bahnhofsbuchhandel), sowie Österreich und Schweiz: Verlagsunion, 6200 Wiesbaden. Anzeigenverwaltung: E. Herwig. Anzeigenpreis: Es gilt die Anzeigenliste Nr. 1. Bitte Media-Unterlagen anfordern. Bezugspreise: Einzelheft (Inland) DM 5,50, Abonnement (Inland) DM 55,-, (Ausland) DM 75,-. Autoren und Manuskripte: Bei Zusendung von Manuskripten und Datenträger erteilt der Autor dem Verlag die Genehmigung zum Abdruck und Versand der veröffentlichten Programme auf Datenträger. Für die mit Namen des Verfassers gekennzeichneten Beiträge übernimmt die Redaktion lediglich die presserechtliche Verantwortung.

Computer-Börse

Verkäufe

Der ganz heiße Börsentip vom eigenen Computer
Stockmaster: grafische **Stock-Control:** mathemat. Auswertung f. Spectrum 48K je Pr. 49,- DM zus. nur 79,- DM.
Töngi
Aspeltstraße 4
6500 Mainz

ZX 81/16 K Bundesliga-Fußballtabelle 1./2. Liga mit Heim-/Auswärtsbilanz Cass. 20,- DM + Listing 5,- DM + Info Gratis + Freiumschlag, H. W. K. Bostel, Friedenstr. 14, 2084 Rellingen 1

Verk. **ZX 81/16 KCMEMO** + 80 Pr. + Reset + Shift Look + Rem-Ansteuerung + gr. Tastatur + Bücher + 90 Listings + Ein/Aus + gr. Netzteil für 350,- DM: Suche Spektrum 48 K, T. Bachmann, Johannisbruch 9, 2357 Springe 1

TI 99/4A + Exbasic + div. Bücher + Rec. + Kabel + Basic-Kurs, VB 750,- DM.
Tel.: 0 28 61 / 48 45
Sa. ab 18.00 Uhr

Verkaufe **ZX-81 + 16 K + Drucker** VB 300,- DM. Suche Atari- 600XL oder C-64. Dirk Wylezol, Am Mühlbachbogen 41 b, 8052 Moosburg

C-64-Superprogramme gegen Unkosten oder Tausch. Gratisliste bei Postlagernd 058542 B, 7900 Ulm/Donau, Diskette oder Kassette

Atari 400/800/600 XL/800 XL Software auf: C. + D. Liste anfordern bei: Power-Game Versand, Oerlinghauser Str. 37 b, 4800 Bielefeld 18

Verkaufe
Anrufbeantworter
Drahtlostelefon
Telefon-Anlage
Eurosignal 1200,– DM
div. USA-Telefone
Info bei:
Reinhard Winner
Höchbergerstraße 62
87 Würzburg

Spielend **Basic lernen** * **Kass.** mit 10 Pr. + Listing + **Erklärung + Tips + Tricks** 20,-.
A. Meuser
Goethestraße 18
5205 St. Augustin 1
ZX Spectrum Super Software

Verkaufe **Taschenrechner TI 59,** Disketten 8 Zoll SS/SD, Preise VHS. Willi Hirschlein, H. v. Hohenlohestr. 44, 6990 Bad Mergentheim, Tel. 0 79 31 / 4 16 05

Suche/Tausche Software für Atari 400/600XL/800XL/vorzugsweise Tausch, Christoph Pieper, Im Fuchsloch 37, 6901 Dossenheim, Tel. 0 62 21 / 86 96 44, schickt mir eure Listen/ ich warte

ZX 81 mit 64 KB, 5 Monate alt, in Philips-Terminal zu verkaufen, als Beigabe 15 Spiele für den ZX 81. Angebote an: M. Kraemer, Weidenweg 6, 4970 Bad Oeynhausen.

Here is Wimpy!
Hello User, biete für C-64 Disk beidseitig mit Topprogrammen für DM 50,- auch Tausch.
Mr. Wimpy
4350 Recklinghausen
Postfach 1017 21

Commodore C 64 nur DM 679,-, Floppy 1541 nur 679,-, Drucker MPS 801 nur DM 669,-. Disketten 10 Stck. 55,-. Alle Geräte neu mit Garantie. Anruf lohnt sich!!! Tel.: 0 75 62 / 34 30!!!

TI 99/4 A Ext.-Spiel nur für Erwachsene – bitte Alter ang. zum absoluten Preishit von nur DM 10,- im Umschl. an T. Karbach, Remscheider Str. 18, 5650 Solingen 1

*** **Apple Schallplatten/MC** – Verwaltungsprogramm für 1 Laufwerk! Suchen: Titel od. Nr. ***600!! Eintr./per Zugr. insg. 1800 Titel! DM 40,- bei VK: H. Lössin, HH 50, Arnoldstr. 48

TI 99/4A. Erstelle Druckerlisting v. Ihren Programmen in TI- u. ext. Basic, je Pgm.-Zeile 4 Pf. + 2,50 DM f. Portokass. m. V.-Scheck o. Briefmarken an H. Schwab, Loch 21, 8504 Stein

Ti-99/4a und alle Atari-Computer für Ti:
Adventure Modul mit Cassette Mission Impossible und Pirate Adventure: 160,- DM für Atari Star Raiders für 70,- DM bei
O. Toffolo
Tel. 0 46 62 / 17 33

Strategie auf Atari. Erobern Sie den Pluto! Spannendes Computergesellschaftsspiel für 2 Sp. Nur 19,50 incl. D. o. C. zzg. Porto + NN. bjh-Vision, Am Sportplatz 6, 2061 Elmenhorst

Ankäufe

TI-99/4A
Suche **Extended Basic Modul** evtl. Tausch gegen **Editoriassembler** sowie Maschinensprachprogramme z. B. **Flugsimulator** und **Ex. Basic Spiel Miner 2049** (USA-Version).
Tel. 0 71 32 / 3 76 08

Apple II suche Programme, Schachspiele, Flug-Simulation, versch. Vorspanntitel, Grafik für Video-Filme, T. Pavone, Heilbronnstr. 6, 7146 Tamm

Suche **Seikosha 6P-100A** Drucker mit Interface für ZX Spectrum, Lothar Schröder, Dellchenstraße 3, 6731 Franckeneck

Suche: Gute und billige Software auf Diskette für 64K Atari 600XL. Angebote an: Michael Ebeling, Riepener Straße 7, 3061 Beckedorf

Suche **Atari Diskettenstation,** Aussehen egal, Hauptsache sie funktioniert. Bitte anrufen unter Berlin 0 30 / 4 93 – 19 27

ZX Spectrum Soft- und Hardware. Preiswert durch Direktimport. Info bei: Ursula Kunz, Junge Hälden 3, 7500 Karlsruhe 41

Suche **Software** für Atari 600XL, zahle gut! Schickt eure Liste an: Mario Noack, Raabestr. 36, 3420 Herzberg

Kontakte

Hallo Spektrumfreaks. Ich möchte mit euch gerne Prog. tauschen. Liste, ob groß oder klein, an: Robert Weidinger, Schönmetzlerstr. 10, 8050 Freising oder Sams.– Diens. 0 81 61 / 43 72

Kontakt mit MZ700-Leuten gesucht. Tel. 05 11 / 42 29 75

***** Atari-Spectrum-User-Club *****
monatl. Info, bundesweit organisiert, riesige Prog.-Bibliothek usw., melden bei:
ASUC, Am Schürhof 17
4905 Spenge
Tel. 0 52 25 / 20 68

Comp.-Club
Bad Hersfeld
sucht Mitglieder!
Kontakt über
Roland Reyer
Am Giegenberg 21
6438 Ludwigsau 1
oder Jugendhaus Hersfeld

Tausch

Tausche **ZX Spectrum-Software** (16/48), auch Microdrive-Software benötigt.
Michael Jedral
Horststraße 6
6729 Leimersheim.
Schreibt schnell. Suche Kopierprogramme für Microdrives

Suche/Tausche **Software** für C 64 und Atari 600/800 XL. Vorzugsweise Tausch.
Oliver Kellner, Haydnstr. 39, 6905 Schriesheim,
Tel. 0 62 23 / 6 59 39

VC 64: Tausche irre Superprogramme, über 40 echte Spielhallen-Games. Liste gegen Rückporto bei Michael Branderberger, Vor der Gasse 82, CH-8200 Schaffhausen

ZX-Spectrum
Software und Interessenaustausch
Monika Baumgartner
4021 Linz
Postfach 142 Austria
Bitte Liste beilegen

Computer-Börse

Verschiedenes

16 K Modul f. ZX 81, 65,- DM. Trak-Ball für Atari + Commodore, 1 Fevertaste, 99,- DM. Beides neu, unbenutzt. Verkaufe (2 DM – 4 DM) + tausche Spectr. Softw. Liste gr. T. Schröer, Hohlstr. 11, 6791 Steinbach

UFO-Forschung. Informationen über UFOs etc. kostenlos. Suche Interessenten zum Aufbau einer Computer-AG (c 64), spez. für wiss. UFO-Forschung! H. W. Peiniger, PF 2361, 5880 Lüdenscheid

VZ 200 Softw. Info g. Freiumschlag. W. Fischer, Friedrichstr. 14, 6463 Freigericht 5

Orig. engl. Software von EMJAY, (Wargame News, The War Machine), für Commodore-Rechner ab 16k und CBM 64! Näheres gegen Freiumschlag von: H. Topf, A.-Bucherer-Straße 63, 5300 Bonn 1

Die richtige Soft-Hardware? Ich sage Ihnen in welcher Zeitschrift. Info gegen frank. Rückumschlag bei: M. Kiwitt, 2000 Hamburg 74, Postlage RND

TI 99/4A Software Service von Action bis Praxis. Info geg. 1. – Rückporto Behringstr. 45, 4600 Dortmund 50

C-64-Synthesizer: ca. 40 K, echte Noten, Partiturediting, Notenausdruck, Direkteinspielung + Notenumsetzung, Supersound. Info von Autor W. Kracht, Espellohweg 38, 2000 Hamburg 52

Roulettesoftware f. Sharp PC 1211/12/4551/PC 1500. Roulettesystemprogramme »Auf Dauer gew.« »AD Astra« »Number one« «Acclaim« »Plein Prive« »Accord« DM 550,- Cass. B. Nowotny, Giselastr. 3, 8000 München 40

An alle SF- und Fantasy-Freunde! Jeweils 8 Mitspieler für Strategiespiele gesucht. Mit Gewinnchancen. Näheres gegen Freiumschlag von: H. Topf, A.-Bucherer-Straße 63, 5300 Bonn 1

ZX-SPECTRUM! Spielend BASIC lernen Ca. mit 10 Pr.+Listing+Erklärung+Tips+Tricks für den ZX. 10,- DM bei A. Meuser, Goethestr. 18, 5205 St. Augustin
NEU - SUPER ZX Spiele -

Spectrum Softw. z. B. M. Miner 8,-; ZOOM 8,-; Liste + kostl. Programm anfordern, Karte genügt: H. Novak, Egerländer Pl. 19, 6053 Obertshausen 2

TI-99/4A: ca. 90 TI-Basic und ca. 80 Ex-Basic Programme zu verk. Info g. Rückumschlag. B. Knedel, Tulpengasse 16, 3171 Weyhausen, Tel. 0 53 62 / 7 11 87



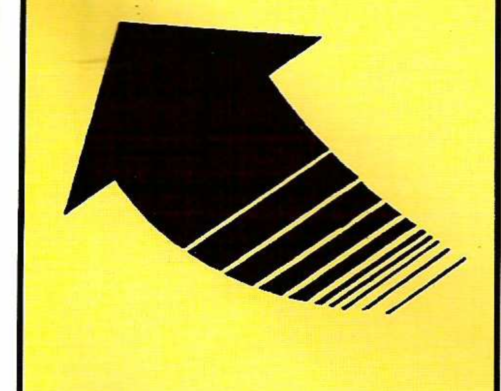
Aktion: Billige Kleinanzeigen
Liebe Leser,
wollen sie ein Kleinanzeige aufgeben? Suchen Sie einen gebrauchten Home-Computer, Software oder Kontakte zu Computer-Clubs etc.?

Dann nutzen Sie jetzt den Kleinanzeigenservice von „Computronic". „Computronic" ist die Zeitschrift für Home-Computer-Besitzer. Sie erreichen durch uns eine Vielzahl von Lesern im In- und Ausland.

Und so wird's gemacht:
Kleinanzeigen-Coupon ausfüllen (alle Zeilen können ausgefüllt werden) und gegen
Vorkasse von DM 4,50 einsenden an:

Tronic-Verlag
Postfach 41
3444 Wehretal 1

Also, schnell ausfüllen und einwerfen in den nächsten Postkasten. Dann ist mit Sicherheit Ihre Kleinanzeige dabei.



Computronic Kleinanzeigen nur gegen Vorkasse

Bitte veröffentlichen Sie ab nächstmöglicher Ausgabe unter der Rubrik:

☐ Verkäufe ☐ Kontakte ☐ Geschäftsempfehlungen
☐ Ankäufe ☐ Verschiedenes ☐ Tausch

folgenden Text:

Bitte jeweils 28 Buchstaben pro Zeile! Bitte Absender nicht vergessen!

Preis für Kleinanzeigen: Private Anzeige DM 4,50 je Zeile
gewerbl. Anzeige DM 8,00 je Zeile
Chiffregebühr DM 10,00 je Anzeige

Name/Vorname:

Straße, Nr.: PLZ/Ort:

Datum, Unterschrift

privat ☐ gewerbl. ☐
(bitte ankreuzen)

Kassettenservice



HEFT 4

VC-64 K = 8,–– DM
D = 15,–– DM
Mauern, Widerstand

ZX-Spectrum K = 12,–– DM
Inventur

TI 99 K = 8,50 DM
Drei-Kronen-Spiel
Zahlenputzen

VC-20 K = 11,50 DM
D = 18,–– DM
Hangman, Saurer Regen,
Quadr. Gleichung

ZX-81 K = 8,50 DM
Invader, Gun-Man

Dragon 32 K = 6,–– DM
Blizzard

Apple II K = 14,50 DM
D = 19,50 DM
Wilder Westen,
Karambolage,
Maskengenerator

Atari K = 10,50 DM
Mastermind,
Schlangenkrieg



HEFT 5

TI 99 K = 14,50 DM
Karl der Käfer
Alien-Landing

VC-64 K = 15,50 DM
D = 23,50 DM
Space-Comets
Erdspalte
Sprite-Data

Apple II K = 9,50 DM
D = 19,50 DM
Musik-Maker
Mission-Adler
Disk-Catalog

Sharp MZ 700 K = 8,50 DM
Kalender
Sharp PC 1500 Lotto

Dragon 32 K = 10,–– DM
Space-Flight, Geosoft

ZX-81 K = 10,–– DM
Go-Ball, Grand-Prix

ZX Spectrum K = 6,50 DM
Missile-Comment

Atari K = 12,50 DM
Tank-Battle
Oil Panic



HEFT 6

VC-64 K = 16,50 DM
D = 23,50 DM
Data Generator
Autostart
Bestellschein
Roadpainter

Dragon 32/64 K = 8,50 DM
Wargames

Apple II K = 12,50 DM
D = 19,50 DM
Snake
Super Datei
Shape-tables

VC-20 K = 8,50 DM
D = 15,00 DM
Bestellschein
Glücky

ZX-81 K = 10,–– DM
Moon-Crash
ZX-Draw

ZX-Spectrum K = 13,50 DM
Defender
Lui der Wurm
Alternativer Zeichensatz

TI-99 K = 14,50 DM
Jack the Digger
Noah – 2099

Atari K = 12,50 DM
Tomstone-City
Schneevogel

Computronic Bestellkarte-Cassettenservice

Alle im Heft abgedruckten Programme können als zusätzlicher Service, zum Selbstkostenpreis, über den Verlag bezogen werden. **(Ausland nur gegen Vorkasse)**

Die Zustellung erfolgt: gegen **Vorkasse** ☐ innerhalb von 1 Woche

oder Inland per **Nachnahme** ☐ + Versandkosten

Achtung: Disketten können vorerst nur für: Apple
VC-64
VC-20 bezogen werden!

Entnehmen Sie bitte aus unserer Preisliste die notwendigen Angaben für Ihre Bestellung:

Bitte liefern Sie mir:

☐ Cassette für ______________ ☐ Anzahl
System

☐ Diskette für ______________ ☐ Anzahl
System

zum Preis von gesamt ______________ DM

Name/Vorname: ______________ Straße, Nr.:

PLZ/Ort: ______________ Datum, Unterschrift

Kassettenservice

Jedes Programm in Computronic wird einer Endkontrolle in unserem Hause unterzogen und genauestens geprüft. Alle im Heft abgedruckten Programme sind nach der richtigen Eingabe der Listings auch funktionsfähig. Viele Leser verlieren jedoch schnell die Geduld am Programmieren, sollte etwas einmal nicht klappen. Die häufigste Ursache von Störungen im Programm, sind unterlaufene Fehler bei der Eingabe. Verzweifeln Sie nicht, sollten Sie einmal keine Zeit zum Programmieren haben oder sollte etwas nicht gelingen. Alle Programme werden im Verlag gespeichert und können jederzeit mit beiliegender Bestellkarte zum Selbstkostenpreis bezogen werden.
Tragen Sie bitte alle notwendigen Angaben in die Bestellkarte ein.

Rückgabe-Garantie:
Wir garantieren:
- kostenlosen Umtausch von defekten bzw. transportgeschädigten Datenträgern!
- die Zusendung der Umtauschware erfolgt **noch am Posteingangstag!**

Bestellung per Telefon:
Wenn es schnell gehen soll ... rufen Sie uns an. Wir nehmen Ihre Bestellung gern entgegen.
Tel.-Nr.: 0 56 51 – 4 06 93 oder **0 56 51 – 4 06 43**
Tronic-Verlag, Postfach 41, 3444 Wehretal 1

Bitte beachten Sie:
Sie ersparen sich zusätzliche Kosten (bis zu DM 5,-), wenn Sie per Vorkasse (bar, Verrechnungsscheck) bestellen.
Ausland: ● Bestellung nur gegen Vorkasse!



VC-64 K = 15,50 DM, D = 19,50 DM
Hardcopy
Space-Fighter
Data-Generator

ZX-81 K = 10,-- DM
Tonprogramm
Aldebaran

Atari K = 12,50 DM
Startup
Zeilen-Split
Chopper-Flight

VC-20 K = 11,-- DM, D = 15,50 DM
Multigraph
All-Rammer

ZX-Spectrum K = 12,50 DM
Matheprogramm
Bongo-Beecatcher

Apple II K = 12,50 DM, D = 19,50 DM
Library
Fight

Dragon 32 K = 8,50 DM
Laser-Attack

TI-99 K = 14,-- DM
Lift Bär
ASC II DEF TEIL 1



C-64 K = 16,50 DM, D = 23,50 DM
Monster Attack
Block-Painter
Epson-Drucker

Atari K = 14,-- DM, D = 19,50 DM
Painter
Hardcopy

Apple II K = 14,-- DM, D = 19,50 DM
Reversal
Disk-Menue-Generator

TI-99 K = 14,50 DM, D = 19,50 DM
Maya
ASC II DEF Teil 2

ASC II DEF Teil 1+2 D = 19,50 DM

Laser 2001 K = 8,50 DM
Andromeda

ZX-81 K = 10,-- DM
Irrgarten 3 D

ZX-Spectrum K = 14,50 DM
Solitaire
Superstat.
Kleinstes gem. Vielfaches

VC-20 K = 11,-- DM, D = 19,50 DM
Zyklo
Meteorit

Dragon K = 13,-- DM
Hardcopy

Tontaubenschießen D = 19,50 DM
Ein Spiel mit sehr guter Graphik für den **C-64**

Im nächsten Heft

Neues, Infos, Tips + Tricks

und natürlich wieder jede Menge Spiel- und Anwendungsprogramme, wie:

Crazy-Cake

wieder ein Klasseprogramm für den *Laser 2001*



Silver-Spar

für den *TI-99* und

Jump

ein Spiel für alle *ZX-Spectrum* Freunde. Mit toller Graphik



5 gewinnt

für den Atari, wo das Programmieren wieder richtig Spaß macht. Lassen Sie sich überraschen. Außerdem bringen wir Programme für: *Colour Genie, Sharp, VC-20, C-64, Dragon 32/64, Apple II* und den *ZX-81*.

Ab 24. 9. 84 bei Ihrem Zeitschriftenhändler.

Es verabschiedet sich:
Die Computronic-Redaktion



computronic

Infos

Tips und Tricks für Einsteiger, Anfänger und Fortgeschrittene. Informationen und Daten. Sinnvolle Anwendungen und Anregungen. Lernen, anhand von Computronic, Programme einzugeben und Programme zu schreiben. Von Basic bis Maschinensprache. Neues vom Computermarkt über Hard- und Software.

computronic

Software-Listings

Unser Schwerpunkt in den Ausgaben (70%). Leichte und anspruchsvolle Programme werden im Heft abgedruckt, wobei die Redaktion alle bekannten Home-Computer berücksichtigen wird. Wenn man will, kann man die Programme einfach eingeben, damit spielen oder arbeiten. Alle Konzepte und Begriffe sowie alle Spiele werden vor ihrer Anwendung erklärt.

computronic

Software-Service

Abgedruckte Programme können als zusätzlicher Service über den Verlag bezogen werden. Wer kein Interesse hat seinen Computer zu »füttern«, oder wem es zu langweilig wird zu programmieren, kann alle abgedruckten interessanten Spiele auf Cassette und Diskette erwerben.

Computronic erscheint monatlich im Tronic-Verlag. Der Bezugspreis beträgt im Inland DM 5,50. Im Preis ist die gesetzliche Mehrwertsteuer enthalten.

Tronic-Verlag GmbH · Postfach 41 · 3444 Wehretal 1 · 05651-40643/40693